

**DISEÑO DE UN ROBOT DELTA PARA EL POSICIONAMIENTO DE
COMPONENTES ELECTRÓNICOS EN CIRCUITOS IMPRESOS DURANTE SU
ENSAMBLAJE**

ARMANDO URIBE CHURTA

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE AUTOMÁTICA Y ELECTRÓNICA
PROGRAMA DE INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2016**

**DISEÑO DE UN ROBOT DELTA PARA EL POSICIONAMIENTO DE
COMPONENTES ELECTRÓNICOS EN CIRCUITOS IMPRESOS DURANTE SU
ENSAMBLAJE**

ARMANDO URIBE CHURTA

**Proyecto de grado para optar al título de
Ingeniero Mecatrónico**

Director

ÁLVARO JOSÉ ROJAS ARCINIEGAS, PhD

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE AUTOMÁTICA Y ELECTRÓNICA
PROGRAMA DE INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2016**

Aprobado por el comité de trabajo de grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar el título de Ingeniero Mecatrónico.:

William Gutiérrez Marroquín

Jurado

José Fernando Gil

Jurado

Santiago de Cali, 26 de septiembre de 2016

A Dios por su infinito amor, por su enorme resguardo y porque me ha brindado todo un universo de bendiciones que me han construido a través de mi carrera académica, llena de experiencias y momentos inverisímiles que han sido de transformación personal, espiritual y profesional.

A mi padre por su inefable respaldo, confianza y acompañamiento durante todo este proceso. A él una dedicatoria especial, pues ha sido la fuente de inspiración del presente proyecto.

A mi madre por darme la vida y permitirme disfrutar de este increíble momento. Son mis padres la razón más grande que tengo para mejorar, crecer y ser mejor, día a día.

A mis hermanos porque son el ejemplo y la motivación que me han permitido mantenerme fiel a mi pasión académica.

A Ángela María por su desmesurado apoyo, a través de quien he podido aprender que las letras son la más hermosa expresión artística del alma.

Son ustedes mi sustento y motivación para culminar y dar lo mejor durante el desarrollo de este proyecto.

El sueño que alguna vez en mí germinó, hoy lo puedo vislumbrar como una verdadera realidad, hecho que no habría sido posible sin ustedes.

Gracias.

AGRADECIMIENTOS

El autor expresa sus más sinceros agradecimientos a:

La Universidad Autónoma de Occidente y la Facultad de Ingeniería, porque cuentan con los mejores docentes en el área de Ingeniería Mecatrónica, de quienes he podido aprender no sólo de sus saberes y conocimientos científicos, sino también de sus innumerables experiencias.

A Álvaro Rojas Arciniegas, director del presente proyecto. Agradecimientos por su apoyo durante el desarrollo de este trabajo.

El Servicio Nacional de Aprendizaje SENA, y el Centro de Electricidad y Automatización industrial CEAI, a través de su coordinador académico César Victoria Cardona, por brindarme su confianza y permitirme compartir el conocimiento a las nuevas generaciones.

A William Gutiérrez Marroquín, profesor, compañero y amigazo. Mis sinceros agradecimientos por su constante apoyo durante todo este proceso académico.

A mi amigo José Fernando Pérez Villa por su valiosa confianza que me ha otorgado aún desde los inicios de mi carrera académica.

A mi amigo Julián Adolfo Cabrera y toda la comunidad educativa del Colegio Nuestra Señora de los Remedios por permitirme crecer bajo una formación en valores y centrada la búsqueda del conocimiento científico. Un agradecimiento especial a Julián por guiarme a través de las mejores sendas del saber.

Mis más sinceros agradecimientos para todos ustedes. Han sido pieza fundamental para mi crecimiento profesional.

CONTENIDO

	Pag.
RESUMEN	17
INTRODUCCIÓN	19
1. PLANTEAMIENTO DEL PROBLEMA	22
2. JUSTIFICACIÓN	23
3. OBJETIVOS	26
3.1. OBJETIVO GENERAL	26
3.2. OBJETIVOS ESPECÍFICOS	26
3.3. ALCANCES Y LIMITACIONES	26
3.4. SIGNIFICADO DEL ESTUDIO Y APLICACIÓN	27
4. METODOLOGÍA	28
5. ANTECEDENTES	30
6. MARCO TEÓRICO	38
6.1. ENSAMBLAJE DE TARJETAS ELECTRÓNICAS	38
6.2. TECNOLOGÍA DE DISPOSITIVOS ELECTRÓNICOS	39
6.3. DISEÑO ELECTRÓNICO (ECAD).	40
6.4. SISTEMA DE COORDENADAS EN PCB.	42
6.5. SISTEMAS DE POSICIONAMIENTO AUTOMÁTICO.	42
6.6. ASPECTOS GENERALES DE LOS ROBOTS MANIPULADORES.	43
6.7. MORFOLOGÍA DE ROBOTS	44
6.7.1. Cartesiano	44
6.7.2. Scara	45

6.7.3.	Articulado (antropomórfico)	46
6.7.4.	Esférico	46
6.7.5.	Cilíndrico	47
6.7.6.	Paralelo	47
6.8.	ASPECTOS GENERALES DE LOS ROBOTS PARALELOS.	48
6.8.1.	Robot de morfología delta.	49
6.9.	MOTORES	50
6.9.1.	Motores AC y DC	51
6.9.2.	Motores paso a paso	51
6.9.3.	Motores sin escobillas	51
6.9.4.	Servomotores	52
6.9.5.	Cinemática de robots	52
6.9.6.	Cinemática directa.	53
6.9.7.	Cinemática inversa.	53
6.10.	GENERACIÓN DE TRAYECTORIAS.	54
7.	DISEÑO CONCEPTUAL	55
7.1.	DESCRIPCIÓN DEL PRODUCTO.	55
7.2.	CONSIDERACIONES DE DISEÑO	55
7.3.	IDENTIFICACIÓN DE NECESIDADES.	56
7.4.	MÉTRICAS.	58
7.5.	CASA DE LAS CALIDADES (QFD).	60
7.6.	ESPECIFICACIONES Y BENCHMARKING	63
7.7.	GENERACIÓN DE CONCEPTOS.	66
7.8.	DISEÑO CONCURRENTE.	69
7.8.1.	Morfología	70
7.8.2.	Motores	71
7.8.3.	Controlador	73
7.8.4.	Software	77
7.8.5.	Mecanismo	80
7.9.	SELECCIÓN DE CONCEPTOS.	85

7.9.1.	Efecto final	86
7.10.	ARQUITECTURA DEL PRODUCTO.	86
8.	DISEÑO DETALLADO	88
8.1.	DISEÑO DE SISTEMA MECÁNICO	89
8.1.1.	Modelo general del robot	92
8.1.2.	Dimensionamiento y espacio de trabajo	98
8.1.3.	Selección de motores	98
8.1.4.	Diseño de base	102
8.1.5.	Diseño de brazos y articulaciones	103
8.1.6.	Diseño del tcp y ventosa.	105
8.1.7.	Soporte estructural	107
8.2.	DISEÑO DE SISTEMA ELECTRÓNICO	109
8.2.1.	Planteamiento	109
8.2.2.	Selección de sistema electrónico embebido	110
8.2.3.	Programación de sistema embebido	112
8.3.	DISEÑO DE SISTEMA DE CONTROL	112
8.3.1.	Geometría del robot	113
8.3.2.	Obtención de la cinemática inversa	114
8.3.3.	Generación de trayectorias	118
8.4.	DISEÑO DE SISTEMA INFORMÁTICO	121
8.4.1.	Requisitos y requerimientos	121
8.4.2.	Selección de plataforma de programación	121
8.4.3.	Interfaz gráfica de usuario (GUI)	121
8.4.4.	Definición de clases, objetos y métodos	126
8.4.5.	Comunicación serial	129
9.	PROTOTIPADO	130

10.	RESULTADOS Y DISCUSIÓN	136
11.	CONCLUSIONES	140
12.	RECOMENDACIONES	142
	BIBLIOGRAFÍA	143
	ANEXOS	146

LISTA DE TABLAS

	Pag.
Tabla 1. Listado de necesidades del proyecto	56
Tabla 2. Ponderación de necesidades (Kano y NUD).	57
Tabla 3. Listado de métricas	58
Tabla 4. Clasificación de riesgos de acuerdo a ISO 13849-1.	59
Tabla 5. QFD Fase 1.	61
Tabla 6. Matriz de correlación del QDF.	62
Tabla 7. Benchmarking de acuerdo a las necesidades (cualitativo).	63
Tabla 8. Relación necesidades y métricas del benchmarking.	65
Tabla 9. Validación selección morfología Delta.	70
Tabla 10. Selección de motores.	73
Tabla 11. Selección del Controlador.	77
Tabla 12. Selección de plataforma de programación.	80
Tabla 13. Evaluación de los conceptos	85
Tabla 14. Dimensiones del robot.	98
Tabla 15. Características del motor seleccionado.	100
Tabla 16. Listado de partes del robot.	131
Tabla 17. Matriz de ensamblaje del robot Delta.	133

LISTA DE FIGURAS

	Pag.
Figura 1. PCB con dispositivos electrónicos tipo “Through Hole”.	19
Figura 2. Izquierda: Diseño de PCB; Derecha: PCB fabricada y ensamblada.	20
Figura 3. Ejemplo de tarjeta electrónica con componentes SMD.	24
Figura 4. Ensambladora SMT Semiautomática.	30
Figura 5. Ensambladora SMT.	31
Figura 6. Rollo de cinta con componentes SMD.	31
Figura 7. Ensambladora SMT - Detalle en compartimento rotatorio de componentes.	32
Figura 8. Ensambladora SMT - Detalle en sistema de sujeción por ventosa.	32
Figura 9. Ensambladora SMT - Vista en planta.	33
Figura 10. Ensambladora de componentes BGA.	34
Figura 11. Ensambladora de componentes SOIC, SOT y QFP.	35
Figura 12. Ensambladora de componentes SMT automática.	36
Figura 13. Ensambladora SMD.	37
Figura 14. Diseño ECAD de Drive para motor DC en Altium.	41
Figura 15. NanoBoard 3000 Series, diseñado en Altium.	41
Figura 16. Robot de configuración cartesiana.	45
Figura 17. Plataforma Stewart-Gough (Seis gados de libertad).	48
Figura 18. Robot Delta (Codian) Pick and Place (Tres grados de libertad).	50
Figura 19. Problema cinemático directo.	53
Figura 20. Problema cinemático inverso.	54
Figura 21. Robot Delta.	55
Figura 22. Caja negra.	67
Figura 23. Descomposición mediante caja transparente.	69
Figura 24. Robot delta con actuadores lineales.	81
Figura 25. Robot Delta con actuadores rotacionales.	82
Figura 26. Concepto 1, mecanismo con actuadores lineales.	83

Figura 27. Concepto 2, mecanismo con actuadores rotacionales y base movil.	84
Figura 28. Concepto 3, mecanismo con actuadores rotacionales y base fija.	85
Figura 29. Arquitectura del robot Delta.	87
Figura 30. Diseño de robot delta.	88
Figura 31. Diseño base del robot.	89
Figura 32. Movimiento sincronizado de motores.	90
Figura 33. Vista en planta del diseño base.	90
Figura 34. Vista frontal del diseño base.	91
Figura 35. Articulación rotacional (1 GDL).	92
Figura 36. Diagrama general del robot.	92
Figura 37. Detalles de la base del robot.	93
Figura 38. Detalle del TCP del robot.	93
Figura 39. Tipos de articulaciones básicas.	95
Figura 40. Eslabones del robot.	96
Figura 41. Articulaciones del robot.	96
Figura 42. Ejemplo de conexión de una red de motores Dynamixel.	99
Figura 43. Motor seleccionado.	101
Figura 44. Motor paso a paso unipolar.	101
Figura 45. Vista isométrica y en explisionado, base superior con motores.	102
Figura 46. Vista superior de la base y medidas	103
Figura 47. Modelo de brazos superiores.	103
Figura 48. Vista superior de brazos.	104
Figura 49. Vista explosionada.	104
Figura 50. Modelo de brazos inferiores.	105
Figura 51. Modelo de la rótula.	105
Figura 52. Modelo del TCP.	106
Figura 53. Vista superior del TCP y medidas.	106
Figura 54. Vista explosionada del TCP.	107

Figura 55. Modelo del Soporte.	107
Figura 56. Vista superior del Soporte.	108
Figura 57. Detalle de zona de sujeción de PCB.	108
Figura 58. Detalle de brazos articulados.	109
Figura 59. Planteamiento del sistema electrónico.	110
Figura 60. Controlador OpenCM9.04	110
Figura 61. Modo de conexión básico hacia motor.	111
Figura 62. Conexión tipo bus mediante puerto serial.	111
Figura 63. Variables geométricas del robot.	113
Figura 64. Interfaces de usuario.	122
Figura 65. Interfaz gráfica de usuario: Inicio de sesión.	122
Figura 66. Interfaz para cambio de contraseñas y administración de usuarios.	123
Figura 67. Interfaz de aplicación principal.	124
Figura 68. Panel de programa.	125
Figura 69. Desglose de piezas del robot.	130
Figura 70. Prototipo del robot desarrollado.	132
Figura 71. Diagrama de ensamble del robot.	134
Figura 72. Prototipo versión 1: Prueba de posición.	136
Figura 73. Prototipo versión 1: Forma de conexión de motores a los brazos.	137
Figura 74. Prototipo versión 1: Ensamblaje de robot en soporte.	137
Figura 75. Prototipo versión 1: Adaptación de motores a brazo.	138
Figura 76. Prototipo versión 1: Vista frontal del robot.	139

LISTA DE ANEXOS

	Pag.
Anexo A. Render del prototipo final del robot.	146
Anexo B. Render del detalle en articulaciones y sujeción de PCB.	147
Anexo C. Código implementado en el sistema embebido electrónico.	148
Anexo D. Especificaciones del motor ServoSmart usados en el robot.	151
Anexo E. Rango de operación del motor ServoSmart.	152
Anexo F. Detalle de pieza conectora entre motor y brazos.	153
Anexo G. Diagrama de ensamble	154
Anexo H. Simulación de movimientos del robot mediante MatLab.	155

GLOSARIO

Articulación: Es una conexión entre dos sólidos que permite que ambos se muevan porque convergen en el mismo eje o punto de apoyo, esto permite el movimiento de ambos en un mismo tiempo con grados de libertad según el diseño específico de cada tipo de articulación.

CAD: *Computer Aided Design* o diseño asistido por computador, es el uso de un amplio rango de herramientas computacionales que asisten a ingenieros, arquitectos y a otros profesionales del diseño en sus respectivas actividades.

Grado de libertad: Corresponde al número de parámetros de entrada que se debe controlar independientemente, con el fin de llevar un mecanismo a una posición en particular.

PCB: *Printed Circuit Board* o circuito impreso, es una placa de material aislante provista de unas pistas o caminos metálicos que sirven para interconectar los diversos componentes que constituyen el circuito en cuestión.

Sistema embebido: Es un sistema electrónico diseñado para realizar pocas funciones en tiempo real, según sea el caso.

SMT: *Surface Mount Technology*, o tecnología de montaje superficial, es el conjunto de procesos usados para soldar componentes de montaje superficial en un circuito impreso.

THT: *Through Hole Technology* o tecnología de agujeros pasantes, es un tipo de tecnología que utiliza los agujeros que se practican en las placas de los circuitos impresos para el montaje de los diferentes elementos electrónicos para crear puentes eléctricos entre una de las caras de la placa de montaje a la otra, mediante un tubo conductor.

TCP: *Tool Center Point* o efector final, es el dispositivo en el extremo de un brazo robótico, diseñado para interactuar con el medio ambiente.

Ventosa: Es un objeto que utiliza la presión negativa del fluido del aire para adherirse a las superficies no porosas.

RESUMEN

En este documento se presenta el diseño de un sistema robotizado tipo paralelo de morfología Delta, el cual es capaz de posicionar componentes de tecnología SMT/SMD dentro de tarjetas electrónicas PCB en el proceso de ensamblaje de las mismas. El proyecto enmarca el diseño de una máquina prototipadora, cuyo público objetivo se encuentra centrado en laboratorios electrónicos dedicados al diseño de circuitos impresos, instituciones académicas, universidades y pequeñas industrias.

Este trabajo se presenta a la Facultad de Ingeniería de la Universidad Autónoma de Occidente de Santiago de Cali, Colombia, bajo la modalidad Proyecto de Grado para optar al título de Ingeniero Mecatrónico. El objetivo principal se encuentra en el diseño de la máquina, detallando su proceso dentro de cuatro objetivos específicos que contemplan todo un desarrollo mecatrónico que se puede resumir en el diseño electrónico, diseño mecánico, diseño de software y el diseño del sistema de control.

En la actualidad, los procesos de elaboración de prototipos de tarjetas electrónicas conllevan diferentes fases en su desarrollo, cuyo proceso implica diferentes formas de ejecución, algunas muy convencionales y/o artesanales, otras muy tecnificadas; sin embargo, la fase de ensamblaje y posicionamiento de los componentes electrónicos, implican en muchos casos una intervención manual debido a la gran diversidad de tipos de dispositivos, formas de ensamblaje y tamaños de componentes.

En los procesos de *prototipado* y ensamblaje, para lograr el posicionamiento y ubicación, se recurre por lo general a procedimientos manuales y semiautomáticos. En el segundo caso, la maquinaria requerida en muchas ocasiones resulta ser bastante costosa, en donde se desarrollan operaciones que requieren directamente de una intervención asistida.

Lo anterior resulta determinante para la elaboración del presente proyecto, el cual plantea el diseño de un robot paralelo tipo Delta, debido a sus grandes prestaciones en velocidad y practicidad en su elaboración. Por otro lado, y teniendo en cuenta que en la actualidad existen gran variedad de aplicaciones informáticas de diseño de circuitos impresos que suministran información como las coordenadas de los componentes, el dimensionamiento y la orientación de los mismos, se ha desarrollado también una aplicación que toma la información en

mención, procesa el posicionamiento mediante el desarrollo cinemático, y ejecuta mediante un sistema de control, las trayectorias necesarias para tomar los dispositivos que estarán ubicados en una zona específica de la máquina y las ubica en el circuito impreso, fijado en la zona central de la máquina. Para la toma y ubicación de los componentes, se ha optado por usar una ventosa neumática.

Palabras clave: E-CAD, PCB, robot, delta, cinemática inversa, SMT, SMD.

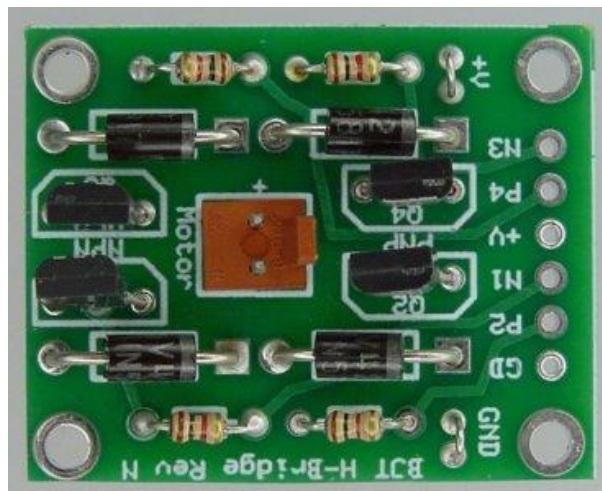
INTRODUCCIÓN

El diseño y elaboración de tarjetas electrónicas es un proceso vital y de suma importancia en la consolidación de un desarrollo electrónico, cuya implementación pretende generar una solución definitiva, plasmada en una placa física, donde una cantidad determinada de componentes electrónicos estarán interconectados con un objetivo específico. En la actualidad, el desarrollo se realiza para dos objetivos particulares: la comercialización mediante producción en masa y el prototipado mediante la producción individual.

Existen compañías dentro del territorio nacional, dedicadas a realizar tarjetas electrónicas, quienes implementan procesos automatizados para su fabricación y ensamblaje. Estas compañías brindan soluciones dirigidas tanto al sector productivo como al sector educativo, sin embargo, al significar costos elevados, muchos usuarios prefieren realizar los procedimientos de manera manual.

La Figura 1 ilustra un producto electrónico donde se tiene una PCB ensamblada con algunos componentes electrónicos básicos, como lo son diodos, resistencias y transistores. Fabricar una tarjeta como ésta puede resultar ser un proceso sencillo, y no se requiere de maquinaria de altas prestaciones o precisión.

Figura 1. PCB con dispositivos electrónicos tipo “Through Hole”.



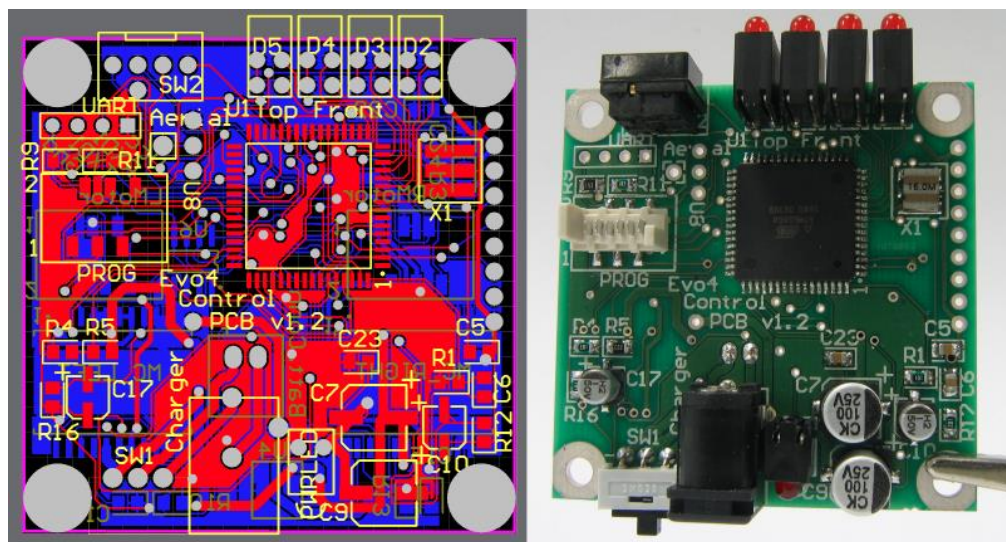
Fuente: RobotRoom, Test Results for an H-Bridge Motor Driver Circuit and GM6 Gearmotor, 2015. [En línea]. RobotRoom, [Consultado Noviembre 2015] Disponible en internet: <http://www.robotroom.com/BipolarHBridge4.html>

Este tipo de desarrollo es muy común y básico, donde los mecanismos de elaboración no exigen un proceso automatizado o máquina para lograr resultados de alta calidad en cortos tiempos de fabricación.

El desarrollo tecnológico acelerado previsto en la actualidad, muestra que hoy en día no basta con desarrollar una tarjeta funcional, pues incluso el medio ambiente ha venido jugando un papel importante. Las condiciones obligan a reducir los consumos eléctricos y mejorar de manera adecuada la transferencia de potencia en los circuitos. Por tales motivos, poco a poco se ha venido recurriendo a la implementación de componentes de menor tamaño, menor consumo y mayor calidad.

Para ello, los desarrolladores electrónicos recurren en sus diseños a la utilización de componentes de bajo consumo de potencia, como lo son los dispositivos de montaje superficial. La Figura 2 ilustra un diseño y la implementación de una PCB, con componentes SMD.

Figura 2. Izquierda: Diseño de PCB; Derecha: PCB fabricada y ensamblada.



Fuente: WARWIK, Warwick Mobile Robotics. [En línea]. WARWIK, [Consultado Noviembre 2015] Disponible en internet: <https://www2.warwick.ac.uk/fac/sci/eng/meng/wmr/projects/mirosot/electronic/>

Se puede observar la gran calidad en el producto terminado, no obstante, se puede analizar que fabricar una tarjeta con esas condiciones, no es tan sencillo como para realizarlo de manera manual.

1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, dentro del territorio nacional, en ambientes no industriales y ambientes académicos, el desarrollo de prototipos de tarjetas electrónicas (PCB) se viene realizando en su gran mayoría con métodos manuales, haciendo que los procesos se tornen engorrosos e ineficientes, acaparando a su vez demasiado tiempo para su elaboración. Esto acompañado de la baja precisión y calidad final en el producto, provoca que el desarrollo de prototipos de PCBs sea un proceso difícil. Cabe destacar que existe gran variedad de métodos para fabricar tarjetas electrónicas, así como para sus etapas intermedias, por ejemplo: ruteo de pistas, elaboración de agujeros, aplicación de anti-solder, aplicación de mascara de componentes, ubicación y posicionamiento de componentes, soldado de componentes, etc.

Cuando se plantea realizar estos procedimientos con componentes de montaje superficial (SMD) y/o componentes de tamaños milimétricos, el proceso se torna más complejo y más aún cuando la cantidad de elementos es considerable. La fase más problemática del proceso está en el posicionamiento y ubicación de los elementos en la tarjeta. Esto genera la necesidad de ¿Cómo desarrollar una máquina de bajo coste, veloz, práctica y de fácil construcción, que permita realizar el posicionamiento de componentes de montaje superficial en un circuito impreso durante el proceso de ensamble?

2. JUSTIFICACIÓN

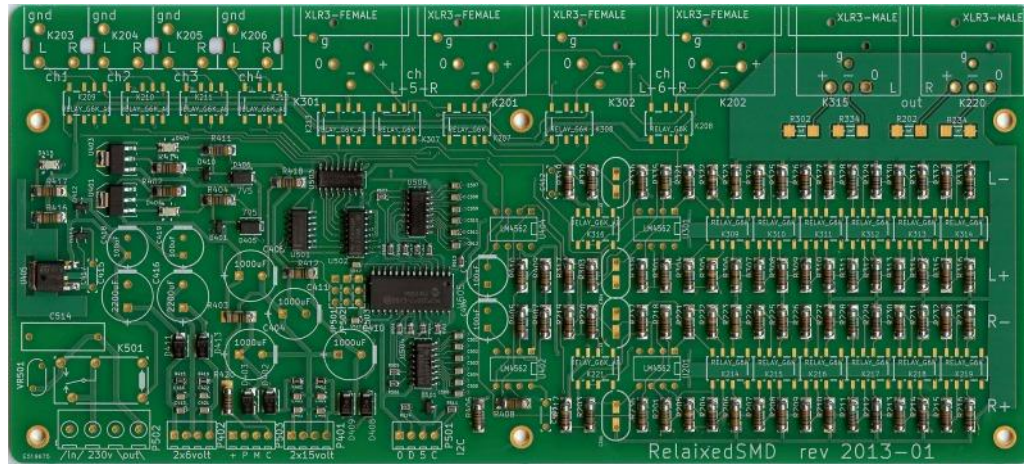
La fabricación y ensamblaje de tarjetas electrónicas es un proceso que se ha venido implementando mediante una vasta cantidad de métodos, tanto manuales como automatizados, pero al momento de analizar dichos procesos dentro ambientes no industriales como pequeñas empresas, centros académicos, institutos y universidades colombianas, se observa que las metodologías implementadas son muy manuales y particulares.

A pesar de ello, los procesos se llevan a cabo, pues en la mayoría de casos académicos, la exigencia en la calidad no es significativa. Sin embargo, cuando se pretende realizar prototipos que involucran componentes electrónicos de tamaños milimétricos, como los componentes SMD (Surface Mount Device) los métodos manuales, sencillos y económicos empiezan a fallar y a generar problemas que causan la frustración de los desarrollos. Esto sucede dado que resulta ser complicado manipular dichos elementos, y más aún cuando se trata de soldar cantidades significativas dentro de tarjetas prefabricadas.

La problemática de fondo no sólo está centrada en ubicar elementos, sino también en el tiempo empleado para llevar a cabo la tarea.

En la Figura 3 se puede ver una tarjeta electrónica con algunos componentes SMD, y se puede observar y deducir a simple vista que manipular y ubicar los elementos de manera manual puede resultar demasiado engorroso.

Figura 3. Ejemplo de tarjeta electrónica con componentes SMD.



Fuente: JOS VAN Eijndhoven, RelaiXedSMD -- DIY balanced pre-amplifier, 2014. [En línea]. eijndhoven [Consultado noviembre 2015] Disponible en internet: <http://www.eijndhoven.net/jos/relaixedsmd/>

Por otro lado, la maquinaria existente que se encarga de realizar dicha labor de manera automática puede ser muy costosa, lo cual disminuye las posibilidades a una institución educativa o pequeña empresa de adquirirla. El objetivo de prototipar tarjetas electrónicas está en el determinar posibles soluciones finales, sin considerar el hecho de comercializar o fabricar en masa, ello representa un paso posterior.

En el presente proyecto, se pretende plasmar el desarrollo de una máquina robotizada de bajo coste, que brinda soluciones prácticas dirigidas al sector educativo y pequeñas empresas que deseen implementar el ensamblaje de tarjetas electrónicas en sus procesos de prototipado dentro de la fase de posicionamiento y ubicación de dispositivos electrónicos de manera autónoma, práctica, rápida, segura y económica.

El proyecto enmarca el diseño de un dispositivo robotizado, el cual debe ser capaz de tomar dispositivos electrónicos (dispuestos en una ubicación específica de la máquina) y ubicarlos dentro de una PCB con una posición y orientación determinada.

La solución a la problemática podrá conseguirse a partir del desarrollo de un robot paralelo de cuatro grados de libertad, dado que ésta configuración permite

conseguir movimientos con rangos cortos, pero a velocidades muy superiores en comparación con un robot antropomórfico, cartesiano, scara, entre otros. El hecho de ser paralelo implica directamente la consecución de velocidades superiores respecto de robots en serie. Para lograr una posición en del área de trabajo establecida, basta con mover los motores (su eje) dentro de un rango inferior a los 360 grados, es decir, los motores no requieren girar más de una vuelta para lograr una posición determinada, motivo por el cual las velocidades son considerablemente altas.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Diseñar una máquina que adopte la forma de Robot Paralelo tipo Delta, que permita tomar mediante una ventosa, componentes electrónicos de tecnología SMD dispuestos en un compartimento de la misma máquina, y de manera automática, logre ubicarlos a partir de las coordenadas obtenidas mediante software de diseño ECAD, dentro de una tarjeta electrónica.

3.2. OBJETIVOS ESPECÍFICOS

Diseñar un sistema electrónico micro-procesado que sea capaz de tomar los datos de los sensores y ejecutar acciones de control sobre los motores.

Diseñar un sistema mecánico que acople los elementos fijos y móviles dentro de una estructura robotizada para que, de manera adecuada, responda a los movimientos deseados.

Diseñar un sistema de control de motores que sea capaz de generar trayectorias de acuerdo a la cinemática (directa/inversa) de la máquina.

Diseñar un sistema informático que capture los datos de las coordenadas y orientación de los componentes, suministradas por software ECAD, para que el sistema electrónico los pueda procesar.

3.3. ALCANCES Y LIMITACIONES

El objetivo del presente proyecto consiste en el diseño completo del robot sin tener en cuenta su implementación o construcción, no obstante, en uno de los posteriores apartados se ilustra el diseño de un prototipo de la máquina, basado en elementos reciclables, de fácil consecución y económicos, donde se ilustran además los planos para su elaboración y esquemas eléctricos, electrónicos y mecánicos.

Con el presente diseño no se pretende generar un producto que compita en el mercado con empresas dedicadas a la producción de circuitos impresos (PCB), sino que se pretende mostrar a las pequeñas empresas que se dedican al desarrollo de sistemas electrónicos, instituciones educativas y universidades, una alternativa de bajo coste que se pueda adaptar en sus líneas de fabricación, y que contemplaría una mejoría radical en las fases de ensamblaje de prototipos.

La resolución de la máquina viene definida directamente por la resolución de los motores (4). Se mostrarán dos opciones: el diseño final y el prototipo, donde en el primer caso se implementan motores de gama superior, esto acompañado de una estructura robusta, mientras que para el prototipo se implementan motores de una gama levemente inferior y una estructura mucho más fácil de fabricar.

Las limitaciones de la máquina se podrían segmentar en dos partes: con respecto a la resolución y con respecto al área de trabajo. Ambas serán estudiadas y finalmente se postula una propuesta que combina estos aspectos con el objetivo de obtener un producto económico, de fácil construcción y capaz de responder a las exigencias del ensamblaje de una PCB moderna, de manera rápida y de calidad.

3.4. SIGNIFICADO DEL ESTUDIO Y APLICACIÓN

El proyecto resume a grandes rasgos la integración de los campos de la ingeniería que constituyen Mecatrónica (Mecánica, Electrónica, Informática y Control), es por ello que los cuatro objetivos específicos se centran en los cuatro componentes en mención.

Esto motiva el hecho de generar un producto netamente mecatrónico, por lo cual, un estudiante de ingeniería mecatrónica podrá conocer la aplicación de cada una de las áreas de la carrera, pasando por todas aquellas asignaturas del núcleo común, hasta llegar a la ingeniería de control, instrumentación, gestión de productos, entre otras.

4. METODOLOGÍA

El proyecto concibe cuatro objetivos específicos, y para dar solución a cada uno de ellos se pretende segmentar el proceso en cuatro etapas: diseño de sistema electrónico; diseño de sistema mecánico; diseño de sistema de control y finalmente diseño de software para el control de la máquina.

El sistema electrónico debe ser un sistema microprocesado donde, mediante un sistema embebido o microcontrolador, se debe gobernar las acciones de control que serán ejecutadas por los motores. También éste deberá ser capaz de tomar los datos de los sensores angulares para de acuerdo a una posición sensada, ubicar los motores en una posición deseada.

En el sistema mecánico, se debe investigar acerca de las posibles configuraciones de ensamblaje de un robot Delta (paralelo) para que, encontrada la mejor solución, se permita realizar un diseño en un software CAD (SolidWorks) para simular el rango de operación, la inercia y otras variables que serán vitales para el buen funcionamiento de la máquina.

El sistema de control debe modelarse mediante software CAE (MatLab) de tal manera que se pueda simular la dinámica y cinemática de la máquina para evaluar la respuesta de los motores, y más específicamente, la respuesta a los movimientos.

Finalmente, se debe implementar el desarrollo de un software que permita conectarse al sistema electrónico, para así poder enviar las coordenadas y las trayectorias a ejecutar.

Cabe destacar que el desarrollo del proyecto no se realiza de manera secuencial o mediante subsistemas independientes, pues el proyecto se basa en la metodología del diseño concurrente donde cada una de las partes son elementos que se fundamentan entre sí para generar conjuntamente soluciones apoyadas en las necesidades, requisitos y requerimientos que se logren determinar en todo el proceso de diseño. Esto además permite integrar los diferentes procesos y subsistemas del mismo, de tal modo que se pueda generar un diseño óptimo de

acuerdo a las necesidades primarias y secundarias, ponderación, importancia y la evaluación de los requerimientos.¹

¹ Herramientas para el ingeniero industrial [En línea] Ingeniería Industrial Online, Ingeniería Concurrente, 2015. [Consultado marzo 2016] Disponible en internet: <http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/procesos-industriales/ingenieria-concurrente/>

5. ANTECEDENTES

Dentro de las investigaciones consultadas, se pudo analizar que la gran mayoría de máquinas prototipadoras comerciales requieren de una operación asistida, razón por la cual son bajas las expectativas de adquisición por parte de las entidades. Por otro lado, las máquinas automáticas son en mayor medida de orden industrial para producción en masa, pero no para prototipado.

A continuación, se muestran diferentes maquinarias, equipos y plataformas de diversas tecnologías y marcas, las cuales son empleadas para realizar tareas de posicionamiento controlado.

- **Sistema de ensamblaje semiautomático para componentes SMT.**

Marca: LPKF Laser & Electronics.

Referencia: ProtoPlace S.

Fecha de desarrollo: 2003.

Esta es una máquina ensambladora de componentes SMT semiautomática, donde la toma y ubicación de componentes se realiza de manera asistida.

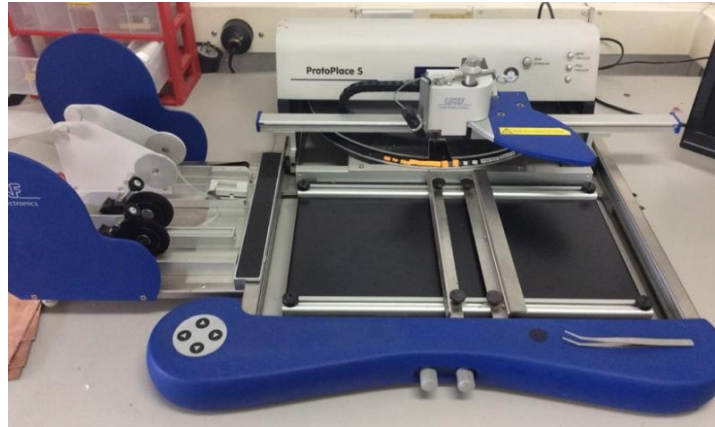
Figura 4. Ensambladora SMT Semiautomática.



Fuente: LPKF Laser & Electronics North America, [En línea]. ProtoPlace S, 2015. [Consultado noviembre 2015] Disponible en internet: http://www.lpkfusa.com/products/pcb_prototyping/smt_assembly/protoplace_s/

Esta máquina es utilizada en el laboratorio de Prototipado del Centro de Electricidad y Automatización Industrial (CEAI) del SENA.

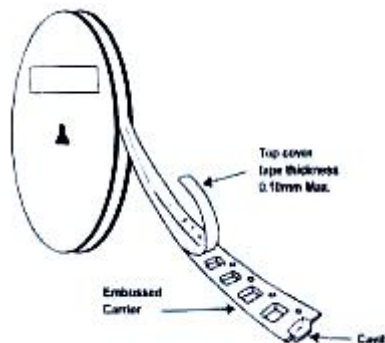
Figura 5. Ensambladora SMT.



Fuente: Laboratorio de Prototipado – SENA, CEAI, 2015.

En la parte superior de la Figura 5, se observa un disco rotatorio. En él se almacenan los componentes electrónicos. Dicho disco posee un motor que gira de acuerdo al compartimento al cual se desea acceder. También posee en la zona lateral izquierda una porta cinta donde es posible disponer componentes electrónicos con distribución en forma continua y encapsulada. La Figura 6 ilustra un ejemplo de cinta.

Figura 6. Rollo de cinta con componentes SMD.



Fuente: SMT Solutions, Tipos de encapsulados SMD, 2015. [En línea]. SMT Solutions, [Consultado noviembre 2015] Disponible en internet: <http://www.smtsolutions.com.ar/encapsulados.htm>

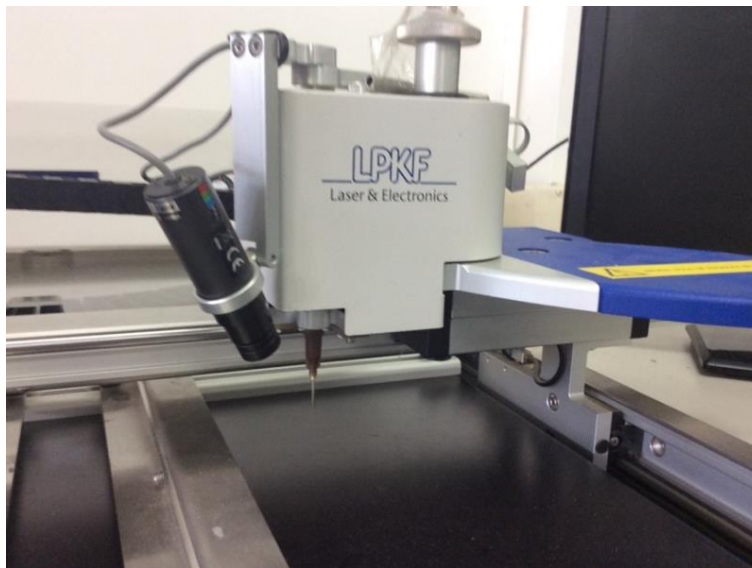
Figura 7. Ensambladora SMT - Detalle en compartimento rotatorio de componentes.



Fuente: Laboratorio de Prototipado – SENA, CEAI, 2015.

Además, cuenta con una cámara VGA que puede ser conectada a un monitor RGB para visualizar el detalle de la sujeción.

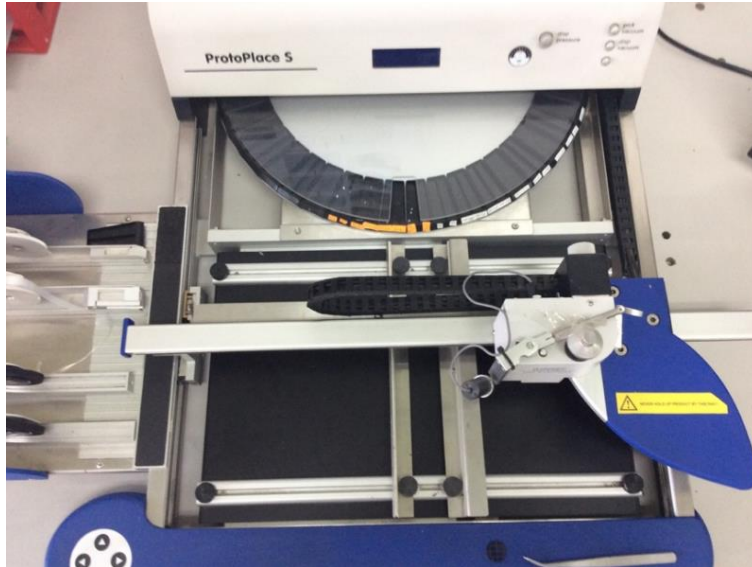
Figura 8. Ensambladora SMT - Detalle en sistema de sujeción por ventosa.



Fuente: Laboratorio de Prototipado – SENA, CEAI, 2015.

La Figura 8 también ilustra el elemento que toma los componentes (ventosa), el cual posee un conducto mediante el cual se toman los componentes en la punta del elemento (tipo aguja).

Figura 9. Ensambladora SMT - Vista en planta.



Fuente: Laboratorio de Prototipado – SENA, CEAI, 2015

- **Posicionadora semiautomática de componentes BGA**

Marca: LPKF Laser & Electronics.

Referencia: ProtoPlace BGA.

Fecha de desarrollo: 2006.

Esta máquina brinda mayor precisión, pero está destinada al ensamblaje de componentes y BGA (Ball Grid Array), componentes de montaje superficial que a diferencia de los dispositivos de tecnología SMT, tienen sus contactos en la zona inferior (Posterior a la zona de la etiqueta). Esta máquina se manipula de manera asistida y cuenta con tres grados de libertad. La movilidad de sus ejes posee un sistema que permite que el movimiento sea más suave que la máquina ProtoPlace S.

Figura 10. Ensambladora de componentes BGA.



Fuente: LPKF Laser & Electronics North America, [En línea]. ProtoPlace S, 2015. [Consultado Noviembre 2015] Disponible en internet: http://www.lpkfusa.com/products/pcb_prototyping/smt_assembly/protoplace_bga/

- **Posicionadora de semi-automatica de componentes QFP, SOIC y SOT.**

Maca: LPKF Laser & Electronics.

Referencia: ProtoPlace E.

Fecha de desarrollo: 1997.

Esta última máquina viene equipada mecánicamente para ser utilizada en aplicaciones para ensamblaje de componentes SOIC (Small Outline Integrated Circuit) y SOT (Small-outline transistor).

Figura 11. Ensambladora de componentes SOIC, SOT y QFP.



Fuente: LPKF Laser & Electronics North America, [En línea]. ProtoPlace E, 2015. [Consultado noviembre 2015] Disponible en internet: http://www.lpkfusa.com/products/pcb_prototyping/smt_assembly/protoplace_e/

Dentro del campo industrial se encuentran muchas máquinas que ensamblan tarjetas electrónicas, sin embargo, la marca en mención se dedica exclusivamente a prototipar. Las soluciones presentadas son semiautomáticas, lo cual requiere que un operador manipule el cabezal de la máquina, y haga el posicionado de los componentes de manera manual.

- **Máquina Pick and Place automática².**

Marca: SMTMax.

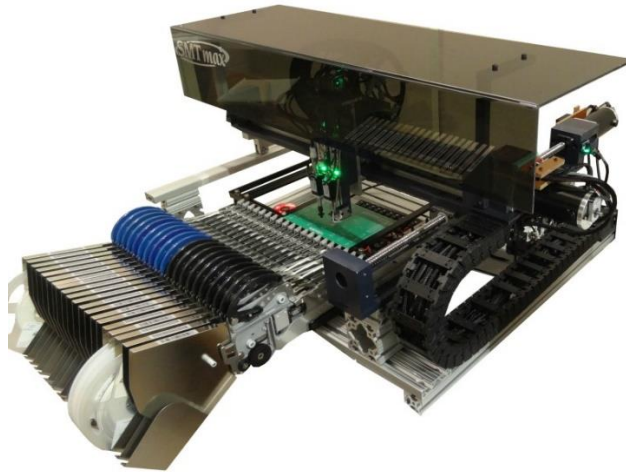
Referencia: QM1100-A.

Fecha de desarrollo: 2002.

La presente es una máquina automática de orden industrial, la más sencilla dentro de la línea que produce la empresa. Es totalmente automática y su uso es destinado a la producción en masa de circuitos impresos PCB.

² Automatic Pick and Place Machines QM1100-A, [En línea] SMTMax, 2016. [Consultado mayo 2016] Disponible en internet: <http://www.smtmax.com/detail.php?id=292>

Figura 12. Ensambladora de componentes SMT automática.



Fuente: SMTMax, Automatic Pick and Place Machines QM1100-A, 2016. [En línea]. SMTMax, [Consultado mayo 2016] Disponible en internet: <http://www.smtmax.com/detail.php?id=292>

El precio de esta línea de maquinarias oscila entre \$100.000 y \$250.000 dólares³.

Entre sus principales características se puede rescatar lo siguiente:

- Two Pick and Place heads
- 3000 estimated pph(parts per hour)
- 13" x 12" working area (QM1100-A)
- 0201 Minimum Component Size
- Computer vision automatic centering system for IC's (SOP, QFP, BGA) and better placement of small parts
- X and Y axis 0.0005" resolution, 0.001" positioning, 0.045° rotational resolution
- Automatic PCB board position correction with fiducial recognition included
- CAD file convert function

(Información tomada del sitio web: <http://www.smtmax.com/>)

³, QM1100 Automatic Pick and Place Machine with Vision system, [En línea] REDDIT 2015. [Consultado mayo 2016] Disponible en internet: https://www.reddit.com/r/PrintedCircuitBoard/comments/358942/qm1100_automatic_pick_and_place_machine_with/ [Consultado mayo 2016]

- **Ensambladora de componentes SMD mediante robot Delta (Experimental)**

Fecha de desarrollo: 2015.

Universidad de Málaga.

El departamento de ingeniería de sistemas y automática de la Universidad de Málaga, desarrolló de manera experimental, una máquina que integra diferentes tecnologías como control numérico y EtherCat Master. La aplicación que se le ha dado corresponde al posicionamiento de componentes electrónicos de montaje superficial. La Figura 13 ilustra el desarrollo realizado en 2015.

De acuerdo a la investigación, el proyecto se ha quedado en un desarrollo netamente académico.

Figura 13. Ensambladora SMD.



Fuente: Departamento de Ingeniería de sistemas y automática, Delta robot + LinuxCNC + Ethercat master - SMD assembly, 2015. [En línea] youtube.. [Consultado Julio 2016] Disponible en internet: <https://www.youtube.com/watch?v=PyXLU3zWdTI>

6. MARCO TEÓRICO

La elaboración del robot Delta implica una serie de procedimientos, cuya base se encuentra sustentada en diversos postulados teóricos. Debido a ello, es fundamental tener claridad sobre algunos conceptos particulares, pues estos se emplearán a lo largo de la pesquisa y arrojarán luces con relación a los aspectos teóricos que integrados podrán establecer la sinergia necesaria para desarrollar el proyecto.

6.1. ENSAMBLAJE DE TARJETAS ELECTRÓNICAS

<Si alguien te habla de una tarjeta gráfica, una placa base o cualquier otro tipo de circuito electrónico, automáticamente pensarás en la placa verde que has visto tantas veces. Y, si eres curioso por naturaleza y no lo sabes, probablemente te harás una pregunta: ¿por qué casi todas son de ese color?>⁴

El desarrollo de la física y la electrónica ha permitido a la humanidad, disfrutar de los más increíbles inventos de la historia. Productos como teléfonos inteligentes, computadoras, controladoras para máquinas industriales, productos quirúrgicos son unos cuantos entre los miles de dispositivos que en su interior poseen como *cerebro* toda una ciudad cuántica. Y es que desde que la revolución electrónica llegó con su exuberante transistor, las cosas en el mundo electrónico cambiaron para siempre.

¿Qué se requiere? La respuesta es simple: interconectar. Un sistema electrónico es una red interconectada de pequeños componentes que cumplen una tarea específica, cuando esto sucede, el producto cumple una labor macro y de utilidad para el consumidor.

Ahora bien, son los circuitos impresos los que han permitido conectar componentes electrónicos dispuestos sobre una base -que por lo general- es de fibra de vidrio. La conexión se realiza gracias a caminos 'dibujados' sobre una placa, generalmente de cobre. Este sistema fue patentado en 1925 por Charles

⁴ Anatomía de un circuito impreso, [línea] En: EL DIARIO, 2014. [Consultado mayo 2014] Disponible en internet: http://www.eldiario.es/hojaderouter/tecnologia/hardware/circuitos_impresos-mascara_de_soldadura-PCB-verde_0_275772820.html

Ducas, sin embargo, su uso y aplicación tuvo que esperar hasta después de la Segunda Guerra Mundial.⁵

En la actualidad, los circuitos impresos o PCB son construidos de diversas maneras, pasando por el popular y colegial método de dibujar con marcador una PCB virgen sometiénola posteriormente al tricloruro de hierro, obteniendo una placa lista para ensamblar, hasta optar por las más sofisticadas metodologías, generalmente implementadas por empresas dedicadas, como lo es el fotograbado, es método screen, entro otros. Donde se obtiene una placa con alta calidad y precisión.

El desarrollo de circuitos impresos es un negocio que en Colombia lo ofrecen empresas como Microcircuitos S.A.; PCB Tecnologías o Colcircuitos.

El ensamblaje de los circuitos impresos es un proceso que por lo general implica una operación manual o asistida debido que existe en la actualidad una vasta diversidad de componentes, como lo son los dispositivos *Through Hole* o los *SMD*; de diferentes formas como los elementos axiales, o radiales, o incluso de diferentes tecnologías como los elementos con encapsulados *DIL*, *Quad-in-line* o *Grid arrays*.

6.2. TECNOLOGÍA DE DISPOSITIVOS ELECTRÓNICOS

Entre las diferentes tecnologías, se destacan dos: la tecnología de agujeros pasantes, y la tecnología de montaje superficial.

La tecnología THT, Through-Hole Technology o tecnología de agujeros pasantes ha sido la más clásica para elaboración de circuitos impresos. De acuerdo con el Instituto para los Circuitos Impresos (IPC) en su estándar IPC-7251, dicha tecnología expone dispositivos electrónicos de gran tamaño, los cuales consumen grandes cantidades de potencia y su instalación dentro de una PCB se realiza mediante inserción. Esta tecnología reemplazó a la antigua tecnología de montaje

⁵ Printed Circuit Board, [En línea] Madehow, 2007, [Consultado enero 2016] Disponible en internet: <http://www.madehow.com/Volume-2/Printed-Circuit-Board.html>

de componentes sobre chasis metálicos, usados en los antiguos televisores y radios.⁶

Por otro lado, el IPC describe una tecnología más reciente, se trata de la SMT, Surface Mount Technology o tecnología de montaje superficial, cuyas raíces datan hacia la mitad del siglo XX. Es una tecnología que aún se mantiene vigente; su uso se ha extendido a la mayoría de aparatos electrónicos de la actualidad debido a las características de los componentes que hacen parte de dicha categoría (SMD) o componentes de montaje superficial. Es una tecnología donde los dispositivos son soldados a la tarjeta electrónica de manera superficial, y no pasante. Las características principales radican en los diminutos tamaños que pueden llegar a tener los componentes.⁷

Ahora bien, conociendo la tecnología existente, es imprescindible determinar su aplicación. Ello será posible a partir de la definición de un diseño electrónico que se adaptará a la misma y potenciará su desarrollo.

6.3. DISEÑO ELECTRÓNICO (ECAD).

El diseño asistido por computador se ha convertido en una herramienta fundamental para el desarrollo de sistemas completos y complejos. La informática y las aplicaciones dedicadas al CAD son esenciales debido principalmente a sus altas capacidades de parametrización.

Existen varias plataformas dedicadas al diseño electrónico asistido por computador o ECAD como lo son: CadSoft EAGLE PCB Design, Altium Designer, OrCAD, PCB Wizard, KiCad, PCB Artist, entre otros.

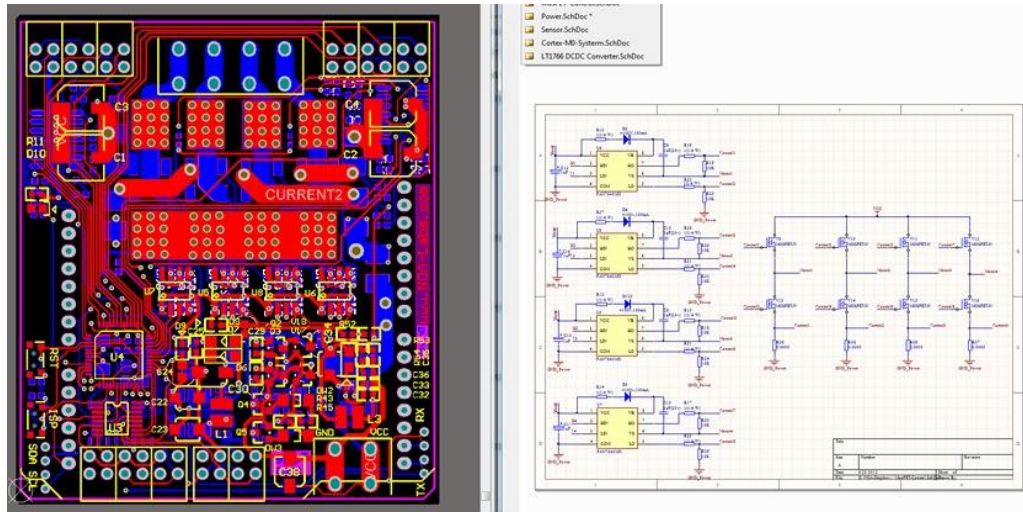
Estas herramientas permiten realizar el conexionado de los diferentes dispositivos activos y pasivos de un circuito y a partir de ello, poder diseñar una PCB. Por ejemplo, una de las herramientas más profesionales de la actualidad es Altium Designer. De acuerdo a esta empresa dedicada al diseño electrónico automatizado, en ella es posible realizar el esquema electrónico y enlazarlo

⁶, Generic Requirements for Through-Hole Design and Land Pattern Standard, [En línea] IPC-7251 [Consultado abril 2016]2008. http://www.ipc.org/committee/drafts/1-13_d_7251WD1.pdf

⁷ Tecnología de Montaje Superficial, [En línea] EcuRed, 2013. [Consultado mayo 2016] Disponible en internet: http://www.ecured.cu/Tecnolog%C3%ADa_de_Montaje_Superficial

directamente a un diseño de PCB, como se ilustra en la siguiente Figura. Incluso también es posible obtener un render 3D de la estética del producto final.⁸

Figura 14. Diseño ECAD de Drive para motor DC en Altium.



Fuente: NOA Labs, Smart Motor Driver, 2015. [En línea]. NOA Labs [Consultado Mayo 2016] Disponible en internet: http://www.noa-labs.com/wp-content/uploads/2013/01/SmartMotorDriverV0.1_AltiumScreenshot.jpg

Figura 15. NanoBoard 3000 Series, diseñado en Altium.



Fuente: Altium, NanoBoard 3000 Series, [En línea]. techdocs.altium. 2016. [Consultado febrero 2016] Disponible en internet: <http://techdocs.altium.com/display/HWARE/NanoBoard+3000+Series>

Después de la concepción del diseño de una PCB, será posible la obtención de un sistema de coordenadas que arroje la ubicación específica que determinará la

⁸ Altium Designer, [En línea] Altium, 2016. [Consultado mayo 2016] Disponible en internet: <http://www.altium.com/>

disposición de los componentes electrónicos dentro de la tarjeta. Por ello es determinante definir este sistema.

6.4. SISTEMA DE COORDENADAS EN PCB.

Algunas de las herramientas ECAD suministran archivos de coordenadas una vez diseñado el PCB. Por ejemplo, EAGLE suministra mediante una ULP, los orígenes de los componentes bajo coordenadas: X, Y, Orientación, esto además de la etiqueta, y el tipo de componente. Un ejemplo del formato:

top,"R1",590.6,196.9,270.0,"R1206"

Siendo:

Top:	La cara en la que se encuentra ubicado el componente.
"R1":	La etiqueta del componente.
590.6,196.9:	Las coordenadas X,Y en mili pulgadas.
270.0:	La orientación en grados.
"R1206":	Resistencia de tamaño 1206.

Esta información será útil para el desarrollo del software de la máquina. Cabe destacar que Altium utiliza el mismo sistema de referencias.

Después de definir el sistema de coordenadas, será pertinente establecer un sistema de posicionamiento automático que permitirá implementar de manera tácita cuál será la tecnología de posicionamiento más adecuada para hacer tangible el desarrollo.

6.5. SISTEMAS DE POSICIONAMIENTO AUTOMÁTICO.

Los sistemas de Control Numérico Computarizado (CNC) han sido trascendentales para la automatización de procesos en donde se requiere manipular, controlar, posicionar o mover elementos. Por ejemplo, máquinas herramientas como fresadoras CNC, aprovechan la tecnología CNC para mover

los cabezales de la máquina y realizar cortes de fresado con precisión y calidad muy superior a lo que sería un control manual.⁹

Por lo general estas máquinas basan sus movimientos en robots de morfología cartesiana debido a la sencillez de su cinemática y facilidad de implementación, generalmente con motores paso a paso.

Al identificar la tecnología que, como se dijo anteriormente, facilitará el desarrollo del diseño, resulta imprescindible mencionar que ésta será utilizada por sistemas robotizados. Tal utilización demanda precisar algunos conceptos concernientes al campo de la robótica.

6.6. ASPECTOS GENERALES DE LOS ROBOTS MANIPULADORES.

Un robot es una máquina que puede programarse para que interactúe con objetos y lograr que imite, en cierta forma, el comportamiento humano o animal.

Según el Oxford English Dictionary, un robot es un aparato Mecánico que se parece y hace el trabajo de un ser humano.¹⁰

Asimismo, se precisan otras definiciones. Michael Brady and Richard Paul (Editores de Robotics Research: The First International Symposium), aseveran que la robótica es la conexión inteligente de la percepción a la acción.¹¹

En este orden de ideas, Joseph L. Jones and Anita M. Flynn (Mobile robots: Inspirations to implementation) han definido que la robótica consiste en el diseño de sistemas, actuadores de locomoción, manipuladores, sistemas de control, sensores, fuentes de energía, software de calidad. Todos estos subsistemas

⁹ AGUILAR SCHAFFER, Julio Alberto Fresado CNC, [En línea] biblio3. 2010, [Consultado abril 2016] Disponible en internet: disponible en: http://biblio3.url.edu.gt/Libros/2013/pro_ma/11.pdf

¹⁰ Robot, [En línea] Oxford Dictionary, 2016. [Consultado marzo 2016] Disponible en internet: <http://www.oxforddictionaries.com/definition/english/robot>

¹¹ BRADY, Michael and Richard Paul, Robotics Research: The First International Symposium 1984.p14

tienen que ser diseñados para trabajar conjuntamente en la consecución de la tarea del robot.¹²

AFNOR (Asociación Francesa de Normalización) adjudican que un manipulador es un mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivos lógicos.

De otra parte, RIA (Robotic Industries Association) afirma que un robot es un manipulador reprogramable y multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especiales a lo largo de trayectorias variables para la realización de tareas variadas (multidisciplinar).

Finalmente, El Instituto Norteamericano de Robótica, describe un robot como un manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas.

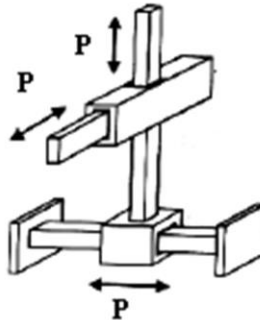
6.7. MORFOLOGÍA DE ROBOTS

Comprendiendo que existe una gama considerable de morfologías de robots, se optará por describir las clases de sistemas robotizados y las diferentes categorías existentes.

6.7.1. Cartesiano. La configuración cartesiana mostrada en la Figura 16 permite que cada una de sus articulaciones se deslice en línea recta a través de los ejes de coordenadas xyz, esta configuración forma un poliedro en su espacio de trabajo.

¹² JOSEPH L. Jones and Anita M. Flynn, Mobile robots: Inspirations to implementation, Segunda edición . Taylor & Francis 1998. P125

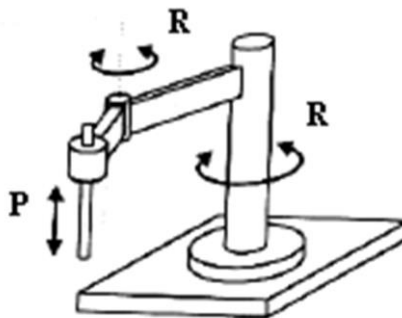
Figura 16. Robot de configuración cartesiana.



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 43.

6.7.2. Scara. Los robots Scara son robots equipados de libertad total de movimientos en los ejes X e Y pero limitados severamente en sus desplazamientos en el eje Z, es decir, se comportan de forma parecida al brazo humano, permitiendo ubicar el extremo de la mano en cualquier ubicación pero siempre sobre el plano. En el eje vertical solo realizan manipulaciones simples que habitualmente consisten en presionar y desplazarse unos pocos centímetros.

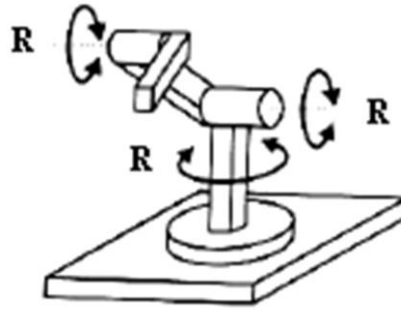
Figura 17. Robot de configuración SCARA



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 42.

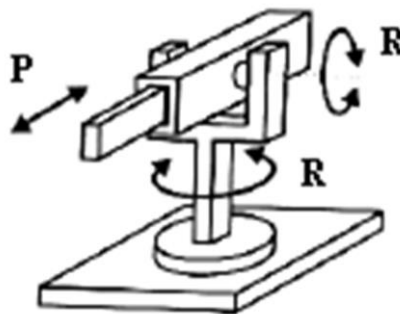
6.7.3. Articulado (antropomórfico). La configuración de tipo antropomórfica se asemeja al brazo humano; por lo tanto, consta de antebrazo y brazo, están conectados a dos articulaciones giratorias denominadas codo y hombro.

Figura 18. Robot de configuración Antropomórfico.



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 39.

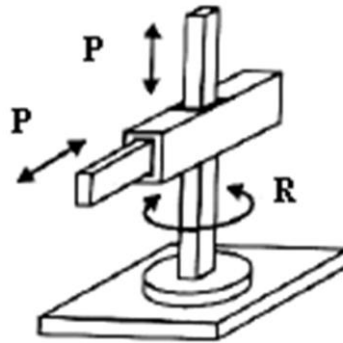
6.7.4. Esférico. En esta configuración, el brazo telescópico se puede elevar o bajar alrededor de un pivote horizontal. El pivote se encuentra sobre una mesa giratoria. Esta combinación de articulaciones le permite al robot desplazar su brazo en un espacio esférico.



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 40.

6.7.5. Cilíndrico. La configuración cilíndrica utiliza un dispositivo deslizante que se mueve a través de una columna, este a su vez se encuentra unido a un dispositivo que le permite realizar un movimiento radial con respecto a la columna. Esta combinación de movimientos genera un espacio de trabajo aproximado a un cilindro.

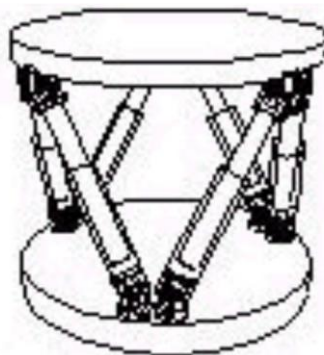
Figura 19. Robot de configuración cilíndrica



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 40.

6.7.6. Paralelo. Los robots paralelos son cadenas cinemáticas cerradas cuyo órgano terminal, o plataforma móvil, está conectado a la base mediante varias cadenas cinemáticas seriales independientes.

Figura 20. Robot de configuración paralela



Fuente: ZOTOVIC Ranko, Principios de la robótica, [Diapositivas], Universidad Politécnica de Valencia, 2014. Diapositiva 44.

6.8. ASPECTOS GENERALES DE LOS ROBOTS PARALELOS.

Un robot paralelo es un mecanismo de lazo cerrado en el cual una plataforma móvil está conectada a una base por al menos dos series de cadenas cinemáticas. Dadas las características de movimiento fino de un mecanismo de lazo cerrado, se pueden encontrar aplicaciones de estos robots en las plataformas de movimiento de los simuladores de vuelo y en aparatos de posicionamiento para herramientas de cirugía de alta precisión.

Recientemente, investigadores están tratando de utilizar estas ventajas para desarrollar máquinas herramienta multi-ejes y herramientas de ensamble de precisión. El ejemplo más claro de este tipo de robots está en la plataforma de Stewart, desarrollada en 1954.¹³

La Figura 21 ilustra una plataforma Stewart-Gough, un robot paralelo de seis grados de libertad.

Figura 21. Plataforma Stewart-Gough (Seis grados de libertad).



Fuente: Robot trepador de estructura paralela, 2017. [En línea]. Universidad Politécnica de Madrid, [Consultado Noviembre 2015] Disponible en internet: <http://sade.disam.etsii.upm.es/trepa/es/project.asp>

¹³ Universidad Autónoma de Bucaramanga, Cinemática inversa y análisis del espacio de trabajo de una plataforma Stewart, [En línea] UNAB, Vol 1, No 1 2009. [Consultado noviembre 2015] Disponible en internet: <http://revistas.unab.edu.co/index.php?journal=mecatronica&page=article&op=viewArticle&path%5B%5D=1242>

Usualmente es implementada en aplicaciones para el desarrollo de simuladores de conducción de automóviles y aviones¹⁴. En los últimos años, se ha venido implementando en aplicaciones para video juegos y también es usada en la NASA para la simulación de sus plataformas aeroespaciales.

Al identificar las características de los robots paralelos, es importante precisar sobre las particularidades del robot de morfología Delta, puesto que éste se enmarca en esta misma categoría.

6.8.1. Robot de morfología delta. La idea básica detrás del diseño de un “Robot Paralelo Delta” es el uso de paralelogramos. Un paralelogramo permite que un eslabón de salida permanezca en orientación fija con respecto a un eslabón de entrada.¹⁵

El uso de tres de estos paralelogramos restringe completamente la orientación de la plataforma móvil que permanece con únicamente 3 grados de libertad. Los eslabones de entrada de estos tres paralelogramos están montados en palancas rotatorias por medio de articulaciones de revolución. Las juntas de revolución pueden ser manipuladas de dos maneras: mediante motores rotacionales o con actuadores lineales.¹⁶

Finalmente, una cuarta pata se utiliza para transmitir movimiento rotacional desde la base al órgano terminal en la plataforma móvil.

¹⁴ Lab of Driving Simulators, [En línea] CEIT, 2014. [Consultado noviembre 2016] Disponible en internet: <http://ceit.es/es/acerca-de-ceit-ik4/departamentos/mecanica-aplicada/laboratorio-de-simuladores-de-conduccion/>

¹⁵ Definición tomada del Capítulo 1 del libro Robótica y Control, del repositorio digital de la Universidad Nacional Autónoma de México.

¹⁶ Ibíd.

Figura 22. Robot Delta (Codian) Pick and Place (Tres grados de libertad).



Fuente: D4 Pick & Place robot, 2015. [En línea]. Codian Robotics, [Consultado diciembre 2015] Disponible en internet: <http://www.codian-robotics.com/en/d4-robots/>

La Figura 22 ilustra un robot Delta de la empresa Codian, de tres grados de libertad, para aplicaciones Pick and Place y soporta cargas de hasta 2kg.¹⁷

Para desarrollar la base funcional del robot, resulta necesario conocer los motores que se pueden emplear, ya que estos permitirán generar los movimientos de un efector final, fundamental para establecer una aplicación específica.

6.9. MOTORES

Un motor eléctrico es una máquina eléctrica que transforma energía eléctrica en energía mecánica a través de interacciones electromagnéticas. A continuación, se describen algunos tipos de motores.

¹⁷, D4 Pick & Place robot, [En línea] Codian 2016. [Consultado marzo 2016] Disponible en internet: http://www.codian-robotics.com/en/robotics/d4_robots/?r=36

6.9.1. Motores AC y DC. Los motores de corriente alterna y los de corriente continua se basan en el mismo principio de funcionamiento, el cual establece que si un conductor por el que circula una corriente eléctrica se encuentra dentro de la acción de un campo magnético, éste tiende a desplazarse perpendicularmente a las líneas de acción del campo magnético.¹⁸

6.9.2. Motores paso a paso. Los motores paso a paso se pueden ver como motores eléctricos sin escobillas. Es común que todos los bobinados del motor sean parte del estator, y el rotor puede ser un imán permanente o, en el caso de los motores de reluctancia variable (que luego describiremos mejor), un cilindro sólido con un mecanizado en forma de dientes (similar a un engranaje), construido con un material magnéticamente "blando" (como el hierro dulce).¹⁹

La conmutación se debe manejar de manera externa con un controlador electrónico y, típicamente, los motores y sus controladores se diseñan de manera que el motor se pueda mantener en una posición fija y también para que se lo pueda hacer girar en un sentido y en el otro.

La mayoría de los motores paso a paso conocidos se pueden hacer avanzar a frecuencias de audio, lo que les permite girar muy velozmente. Con un controlador apropiado, se los puede hacer arrancar y detenerse en un instante en posiciones controladas.

6.9.3. Motores sin escobillas. Los motores Brushless están compuestos por una parte móvil que es el rotor, que es donde se encuentran los imanes permanentes, y una parte fija, denominada estator o carcasa, sobre la cual van dispuestos los bobinados de hilo conductor.

Como su propio nombre indica, Brushless quiere decir "sin escobillas". En este tipo de motor la corriente eléctrica pasa directamente por los bobinados del estator o carcasa, por lo tanto, aquí no son necesarias ni las escobillas ni el colector que se utilizan en los Brushed (con escobillas). Esta corriente eléctrica genera un campo electromagnético que interacciona con el campo magnético creado por los

¹⁸ LÓPEZ ARENALES, José Carlos [En línea] Proyectos de Ingeniería Mecánica, 2012. [Consultado noviembre de 2015] Disponible en internet: <http://biblio3.url.edu.gt/Libros/2013/ing/pim/12.pdf>

¹⁹ CARLETTI, Eduardo J Motores paso a paso: Características básicas, [En línea] Robots Argentina 2014. [Consultado noviembre 2015] Disponible en internet: http://robots-argentina.com.ar/MotorPP_basico.htm

imanes permanentes del rotor, haciendo que aparezca una fuerza que hace girar al rotor y por lo tanto al eje del motor.²⁰

6.9.4. Servomotores. Los servos son un tipo especial de motor de corriente continua que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su intervalo de operación. El servo es, además, un potente dispositivo que dispone en su interior un motor con un reductor de velocidad y multiplicador de fuerza, también dispone de un circuito que controla el sistema. El ángulo de giro del eje puede estar limitado a 180° o 360°, y existen servomotores que permiten el giro continuo. La característica especial de los servos es que en su interior cuentan con un sistema de control para posición o velocidad en los casos más básicos, y en motores más sofisticados, permiten tener un control supervisado del torque, la temperatura, entre otros.²¹

Conocer los tipos de motores es imprescindible para llevar a cabo la implementación funcional del robot; luego será necesario detallar el modelamiento matemático que permitirá describir el movimiento de cada uno de estos.

6.9.5. Cinemática de robots. La cinemática de robots estudia su movimiento con respecto a un sistema de referencia. Además, permite realizar una descripción analítica del movimiento espacial en función del tiempo y relacionar la localización del extremo del robot con sus articulaciones. Esta permite, además, generar una descripción analítica del movimiento espacial en función del tiempo y establece una relación entre la localización del extremo del robot y los valores articulares. existen dos maneras de modelar la cinemática de un robot: mediante métodos directos y métodos inversos.²²

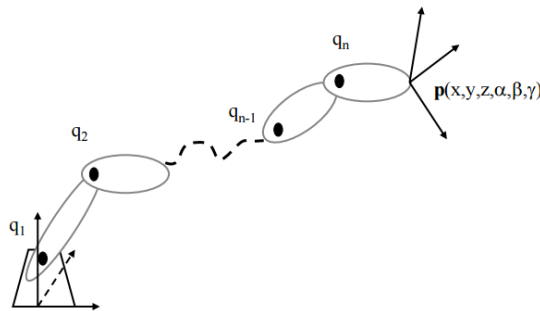
²⁰ BUENO, Diego . Motor eléctrico brushless: Funcionamiento y características, [En línea] 1mecanizadoelarenal 2013. [Consultado abril 2016] Disponible en internet: <https://1mecanizadoelarenal.files.wordpress.com/2013/11/motores-brushless.pdf>

²¹ Servomotores, [En línea] Universidad Técnica Federico Santa María, 2003. [Consultado abril 2016] Disponible en internet: <http://www2.elo.utfsm.cl/~mineducagv/docs/ListaDetalladadeModulos/servos.pdf>

²² Geometría y Cinemática de robots, [En línea] Universidad de Sevilla, 2008. [Consultado febrero 2016] Disponible en internet: http://www.esi2.us.es/~vivas/ayr2iaei/CIN_ROB.pdf

6.9.6. Cinemática directa. La cinemática directa consiste en determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot. Ésta arroja como resultado, las ecuaciones que determinan la posición del efector final (p , ver Figura 23) en función de los valores articulares (q_n , ver Figura 23). Dado que un robot puede considerarse como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia.²³

Figura 23. Problema cinemático directo.



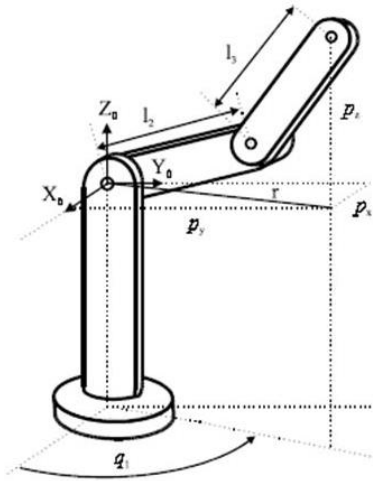
Fuente: RAMÍREZ BENAVIDES, Cinemática Directa del Robot, 2014. [En línea]. Robótica: Kryscia [Consultado Julio 2016] Disponible en internet: <http://www.kramirez.net/Robotica/Material/Presentaciones/CinemáticaDirectaRobot.pdf>

6.9.7. Cinemática inversa. La cinemática inversa consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot para que su extremo se posicione y oriente según una determinada localización espacial. Al contrario que el problema cinemático directo, el cálculo de la cinemática inversa no es sencillo, ya que consiste en la resolución de una serie de ecuaciones fuertemente dependiente de la configuración del robot. La solución a la cinemática inversa depende de la configuración del robot, por lo cual, pueden existir múltiples soluciones. Por ejemplo, si un robot posee más grados de movilidad que grados de libertad, existirán entonces diferentes soluciones, pues de esta manera, el robot podría llegar a una posición dada, de diferentes maneras.

Las ecuaciones resultantes son función de las características geométricas del robot, y el resultado corresponde a los valores articulares (ver Figura 24).

²³ Ibíd. Disponible en internet: http://www.esi2.us.es/~vivas/ayr2iaei/CIN_ROB.pdf

Figura 24. Problema cinemático inverso.



Fuente: RAMÍREZ BENAVIDES, Cinemática Directa del Robot, 2014. [En línea]. Robótica: Kryscia [Consultado Julio 2016] Disponible en internet: <http://www.kramirez.net/Robotica/Material/Presentaciones/CinemáticaDirectaRobot.pdf>

Una vez obtenida la cinemática del robot, es necesario considerar que no basta con adquirir los puntos específicos a donde el robot debe llegar, sino que el recorrido o trayectoria del efector final, resulta significativo al momento de estudiar el movimiento y sus implicaciones dinámicas.

6.10. GENERACIÓN DE TRAYECTORIAS.

Dentro de la robótica, un aspecto de suma importancia está en la generación de las trayectorias que realiza el efector final, pues además de mover el robot hacia una posición deseada, es importante llevarlo bajo unas condiciones ideales para que los aspectos dinámicos del robot como la inercia, no afecten el movimiento.

Por tanto, existen diferentes formas de generar las trayectorias, entre las cuales se destacan las interpolaciones cúbicas, interpolaciones de orden 5 o 7, trayectorias trapezoidales, trapezoidales en S, sinodales, entre otras. El objetivo principal está en conseguir movimientos suavizados, controlando la posición y sus derivadas: la velocidad, aceleración y en algunos casos, la tercera derivada: el jerk.

7. DISEÑO CONCEPTUAL

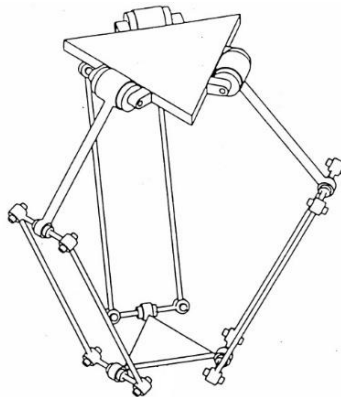
7.1. DESCRIPCIÓN DEL PRODUCTO.

El presente proyecto contempla el desarrollo de un producto robotizado de morfología Delta, de bajo coste, dirigido a pequeñas industrias, instituciones educativas y universidades que requieran implementar en sus procesos de desarrollo de PCB una máquina que permita ensamblar componentes electrónicos de tecnología de montaje superficial de tamaño mínimo 1x1 milímetros, en circuitos impresos de tamaño máximo 20x20 centímetros. El producto incluye: Estructura completa con motores articulados, ventosa, sistema electrónico embebido y software de control desarrollado en lenguaje Java.

7.2. CONSIDERACIONES DE DISEÑO

El diseño del robot se basa en tres cadenas cinemáticas que se acoplan en una base superior y un efector, adoptando la forma que se ilustra en la Figura 25.

Figura 25. Robot Delta.



Fuente: JINBO WU and Zhouping Yin, A Novel 4-DOF Parallel Manipulator H4, 2018. [En línea]. Intechopen. [Consultado mayo 2015] Disponible en internet: http://www.intechopen.com/books/parallel_manipulators_towards_new_applications/a_novel_4-dof_parallel_manipulator_h4

La base superior se debe acoplar a las articulaciones de manera directa, por lo cual, la resolución de los motores definirá la resolución del robot. Las

articulaciones superiores deben acoplarse al efector final mediante tres paralelogramos, uno en cada articulación. Esto, mediante rotulas de dos grados de libertad cada una. Los paralelogramos permiten al robot poder ejecutar trayectorias manteniendo siempre alineado horizontal el efector final. El robot posee cuatro grados de movilidad: tres suministrados por cada motor de las articulaciones, y uno que se conectará en el efector final para poder rotar la ventosa. En apartados posteriores se observa que el robot tiene cuatro grados de libertad, por lo cual el robot es holónimo.

7.3. IDENTIFICACIÓN DE NECESIDADES.

A continuación, se desglosa el listado de necesidades identificadas para el diseño del robot. Las necesidades se identificaron teniendo en cuenta lo analizado en el planteamiento del problema, la justificación del proyecto, los objetos y se toma como referente las prestaciones de la máquina prototipadora semiautomática ProtoPlace S, presente en el Centro de electricidad y automatización industrial del SENA, Cali. También se tuvo en cuenta el tamaño de los componentes a manipular y el área de trabajo establecido (20x20cm).

Tabla 1. Listado de necesidades del proyecto

Número	Necesidad	
1	El robot Delta	es económico.
2	El robot Delta	es preciso.
3	El robot Delta	es exacto.
4	El robot Delta	es seguro.
5	El robot Delta	es de fácil operación.
6	El robot Delta	es de fácil instalación.
7	El robot Delta	es de fácil ensamblaje.
8	El robot Delta	es de fácil mantenimiento.
9	El robot Delta	es robusto.
10	El robot Delta	es de fácil portabilidad.

Tabla 2.(continuación)

11	El robot Delta	es escalable.
12	El robot Delta	permite sujetar componentes de tamaño milimétrico.
13	El robot Delta	tiene un volumen de trabajo amplio.
14	El robot Delta	se puede calibrar fácilmente.
15	El robot Delta	tiene alta resolución.
16	El robot Delta	es veloz

En este orden de ideas, siguiendo el modelo de Kano (Delighters, Satisfiers y Must have) y NUD (Nuevo, Único y Difícil), se asigna una valoración de acuerdo a la importancia, priorizando finalmente las necesidades. Esta información servirá como base para el diseño conceptual del robot.²⁴

Tabla 3. Ponderación de necesidades (Kano y NUD).

Num	Necesidad		Peso	Kano	NUD
1	El robot delta	es económico.	1	S	
2	El robot delta	es preciso.	9	M	
3	El robot delta	es exacto.	9	M	
4	El robot delta	es seguro.	3	M	
5	El robot delta	es de fácil operación.	3	S	
6	El robot delta	es de fácil instalación.	3	S	
7	El robot delta	es de fácil ensamblaje.	3	S	
8	El robot delta	es de fácil mantenimiento.	3	S	
9	El robot delta	es robusto.	3	S	
10	El robot delta	es de fácil portabilidad.	1	D	

²⁴ ROJAS, Alvaro Jose. DISEÑO MECATRONICO: Planeación de productos - identificación de necesidades [Diapositivas], Santiago de Cali: Universidad Autónoma de Occidente, 2014. Diapositiva 35.

Tabla 4.(continuación)

11	El robot delta	es escalable.	1	D	
12	El robot delta	permite sujetar componentes de tamaño milimétrico.	9	M	Di
13	El robot delta	tiene un volumen de trabajo amplio.	1	D	Di
14	El robot delta	se puede calibrar fácilmente.	3	D	
15	El robot delta	tiene alta resolución.	9	M	Di
16	El robot Delta	es veloz	3	S	
N – Nuevo. U – Único. Di – Difícil. M – Must have S – Satisfiers. D – Delighters.					

De acuerdo a la tabla anterior, las necesidades más críticas son la 2, 3, 12 y 15 (Es preciso, es exacto, sujeción de componentes de tamaño milimétrico y alta resolución). Criterios que vienen relacionados directamente con la selección de los motores y el diseño del efector final, por tanto, el proyecto debe asegurar que dichas necesidades puedan ser suplidas mediante un diseño óptimo.

7.4. MÉTRICAS.

Obtenidas las necesidades y su ponderación respectiva, se procede con la realización del listado de métricas relacionadas con los conceptos de diseño y necesidades. Estas métricas permiten medir las necesidades establecidas en la tabla 1.

Tabla 5. Listado de métricas

Número de métrica	Número de necesidad	Métrica	Importancia	Unidad
1	1	Costo	1	\$
2	15, 12	Resolución	9	µm

Tabla 3.(continuación)

3	2, 12	Precisión	9	%
4	3, 12	Exactitud	9	%
5	9, 13	Alto	3	mm
6	9, 13	Ancho	3	mm
7	9, 13	Largo	3	mm
8	13	Volumen de trabajo	3	mm*mm*mm
9	4	Seguridad	3	Tabla
10	9	Impacto visual	1	Subjetivo
11	8	Facilidad de mantenimiento	1	%
12	16	Velocidad	3	mm/s
13	9	Peso	1	Kg
14	11	Escalabilidad	1	Subjetivo
15	7	Tiempo de ensamblaje	1	s
16	5	Facilidad de uso	3	%

Como se mencionó en el listado de métricas, la seguridad debe evaluarse mediante tabla de acuerdo al estándar ISO 13849-1. A continuación, se presenta la tabla de riesgos.²⁵

Tabla 6. Clasificación de riesgos de acuerdo a ISO 13849-1.

Tipo de riesgo	Descripción
A	No eléctrico
B	Electromecánico

²⁵ International Organization for Standardization. Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design, Gennève, Suiza: ISO: 2006. P125

Tabla 4.(continuación)

C	Electrónica compleja
D	A con B
E	C con B
F	C con A o A con B

Fuente: International Organization for Standardization. Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design, Ginebra, Suiza: ISO: 2006. P125

El impacto visual se debe evaluar de acuerdo a la calidad visual de cada uno de los componentes, los materiales a usar y su color, la forma del diseño y las curvas de las piezas.

La escalabilidad se evalúa con respecto a la facilidad que implicaría el aumentar un volumen de producción con respecto al re cambio en el diseño teniendo en cuenta la cantidad de componentes específicos.

El mantenimiento se evalúa teniendo en cuenta el tipo de los materiales a usar y sobre todo aquellos componentes que tienen relación mecánica directa, o elementos articulares que tienen conexión física.

7.5. CASA DE LAS CALIDADES (QFD).

Ahora bien, con el listado de necesidades, su ponderación y el listado de métricas se procede a desarrollar la casa de las calidades o QFD (Quality Function Deployment), a través del cual se puede hacer una evaluación que arroje como resultado un nivel de cumplimiento de las necesidades planteadas con respecto a sus respectivas métricas.

Tabla 7. QFD Fase 1.

No.	NECESIDADES	PESO	Costo	Resolución	Precisión	Exactitud	Alto	Ancho	Largo	Volumen de trabajo	Seguridad	Impacto visual	Facilidad de mantenimiento	Velocidad	Peso	Escalabilidad	Tiempo de ensamble	Facilidad de uso
1	es económico.	1	1															
2	es preciso.	9			9													
3	es exacto.	9				9												
4	es seguro.	3									3							
5	es de fácil operación.	3																3
6	es de fácil instalación.	3																
7	es de fácil ensamble.	3															1	
8	es de fácil mantenimiento	3											1					
9	es robusto.	3					3	3	3			1			1			
10	es de fácil portabilidad.	1																
11	es escalable.	1														1		
12	permite sujetar componentes de tamaño milimétrico.	9		1	1	1	1	1	1									
13	tiene un volumen de trabajo amplio.	1								3								
14	se puede calibrar fácilmente.	3																
15	tiene alta resolución.	9		9														
16	es veloz	3												3				
Prioridad			1	90	90	90	18	18	18	3	9	3	3	9	3	1	3	9
Prioridad Elevada			0,3%	24,5 %	24,5 %	24,5 %	4,9%	4,9%	4,9%	0,8%	2,4%	0,8%	0,8%	2,4%	0,8%	0,3%	0,8%	2,4%

Con lo anterior, se evidencia la importancia de las métricas 2, 3 y 4 correspondientes a la resolución, precisión y exactitud, variables de vital importancia para el diseño detallado.

Teniendo en cuenta esto, se procede a realizar la matriz de correlación (que forma el techo de la casa), en donde se valoran las métricas con el objetivo de buscar contrariedades o implicaciones bien sea positivas o negativas que puedan determinar inconvenientes en el diseño final del producto.

Tabla 8. Matriz de correlación del QDF.

1	Costo																
2	Resolución	-															
3	Precisión		++														
4	Exactitud		++	+													
5	Alto																
6	Ancho																
7	Largo																
8	Volumen de trabajo					+	+	+									
9	Seguridad																
10	Impacto visual	-															
11	Facilidad de mantenimiento	-															
12	Velocidad		+														
13	Peso					+	+	+		-							
14	Escalabilidad								+								
15	Tiempo de ensamblaje													-			
16	Facilidad de uso										+						
Preferencia		◇	▲	▲	▲	◇	◇	◇	◇	◇	◇	◇	▲	▼	◇	▼	▼
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Aumentar	▲
En un valor	◇
Disminuir	▼
Preferencia	

Muy Pos	++
Pos	+
Neg	-
Muy Neg	--
Correlación	

7.6. ESPECIFICACIONES Y BENCHMARKING

Con el objetivo de poner en comparación el robot Delta frente diferentes sistemas que cumplen una función similar a la presente, se realiza un sondeo entre 3 diferentes marcas comerciales, destacando así los aspectos más preponderantes. Los datos en comparación serán relacionados con las necesidades y métricas establecidas con anterioridad, obteniendo así las especificaciones del producto.

Las máquinas que serán puestas en comparación son la ProtoPlace S, ProtoPlace BGA y SMTMax QM1100-A, analizadas en los antecedentes. Las tres se caracterizan por su sistema de sujeción y agarre, sin embargo, una de ellas es totalmente automatizada. Un aspecto importante que se debe destacar está en que la tercera máquina en mención tiene una finalidad más industrial, mientras que las dos primeras son para prototipado.

Otro aspecto importante a tener en cuenta es que las tres máquinas analizadas son de configuración cartesiana, y esto debido a que en la investigación realizada, no se encontró máquinas comerciales prototipadoras de configuración Delta para el ensamblaje de componentes de montaje superficial, por lo cual, el análisis de las especificaciones del producto y el benchmarking se realiza teniendo en cuenta tres de las máquinas analizadas en el apartado de antecedentes.

En la siguiente tabla se realiza la comparación cualitativa, siendo • correspondiente a la relación menos fuerte, y •••• correspondiente a la relación más fuerte de acuerdo a las necesidades.

Tabla 9. Benchmarking de acuerdo a las necesidades (cualitativo).

NECESIDAD	ProtoPlace S	ProtoPlace BGA	SMTMax
es económico.	••	•••	NO
es preciso.	••	•••	•••••
es exacto.	••	•••	•••••
es seguro.	•••••	•••••	••••

Tabla 10.(continuación)

es de fácil operación.
es de fácil instalación.
es de fácil ensamblaje.
es de fácil mantenimiento.
es robusto.
es de fácil portabilidad.
es escalable.	NO
permite sujetar componentes de tamaño milimétrico.
tiene un volumen de trabajo amplio.
se puede calibrar fácilmente.
permite obtener grandes resoluciones.

De acuerdo a la Tabla 7, la máquina **SMTMax** brinda las mejores prestaciones en relación con las necesidades planteadas.

Ahora bien, para ponderar las necesidades de los productos del benchmarking, se relacionan con las métricas listadas en la tabla 3, ilustrado en la siguiente tabla.

Tabla 11. Relación necesidades y métricas del benchmarking.

MÉTRICA	Importancia	Unidad	ProtoPlace S	ProtoPlace BGA	SMTMax
Costo	0,30%	\$	\$15.000	\$18.000	\$140.000
Resolución	24,50%	µm	-	-	88
Precisión	24,50%	%	0.2	0.2	0.008
Exactitud	24,50%	%	0.1	0.1	0.01
Alto	4,90%	mm	180	160	330
Ancho	4,90%	mm	570	520	450
Largo	4,90%	mm	610	540	605
Volumen de trabajo	0,80%	mm*mm*mm	250*250*15	210*210*15	300*300*20
Seguridad	2,40%	Tabla	-	-	-
Impacto visual	0,80%	Subjetivo	-	-	-
Facilidad de mantenimiento	0,80%	%	-	-	-
Velocidad	2,40%	mm/s	-	-	
Peso	0,80%	Kg	7.5	6.4	8.1
Escalabilidad	0,30%	Subjetivo	-	-	-
Tiempo de ensamblaje	0,80%	s	-	-	-
Facilidad de uso	2,40%	%	-	-	-

Como se había mencionado, la máquina **SMTMax** puede brindar altas prestaciones de acuerdo a las necesidades y métricas planteadas, sin embargo, la tabla 8 muestra que el costo de ésta es bastante elevado respecto de las otras dos, por lo cual, dentro del proceso de diseño se hará énfasis en la utilización de materiales y elementos de bajo coste, logrando así un dispositivo de calidad, alta precisión y bajo coste.

7.7. GENERACIÓN DE CONCEPTOS.

La generación de conceptos se realiza con el fin de establecer soluciones al proyecto. En esta etapa se procede a realizar un análisis y una descomposición de las funciones y sub-funciones identificadas. De esta manera se pretende llegar a posibles soluciones en todas las partes y funciones del dispositivo, y así poder determinar la solución más apropiada para cumplir con las necesidades presentadas por el cliente.²⁶

Para empezar, se detalla a continuación la caja negra, donde se observan las entradas y salidas del sistema.

- **Entradas**

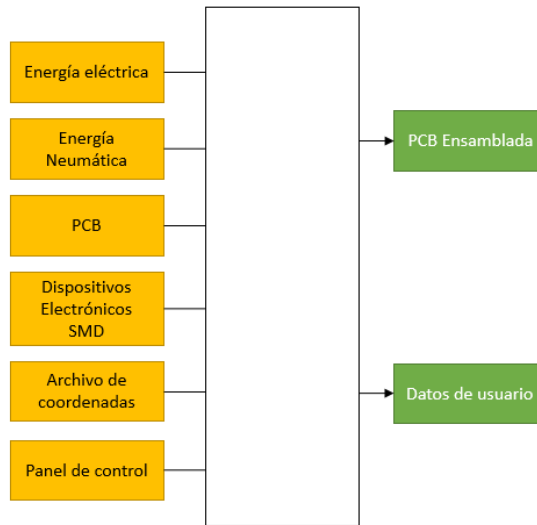
- Energía eléctrica
- Energía neumática
- Componentes a ensamblar (PCB y Dispositivos Electrónicos SMD)
- Archivo de coordenadas
- Panel de control (Opciones de usuario)

- **Salidas**

- PCB Ensamblada
- Datos de estado

²⁶ ROJAS, Alvaro Jose. DISEÑO MECATRONICO: Generación de Conceptos - Ingeniería Inversa [Diapositivas], Santiago de Cali: Universidad Autónoma de Occidente, 2014. Diapositiva 7.

Figura 26. Caja negra.



Fuente: Elaboración propia.

Una vez identificadas las entradas y salidas, se procede a detallar los elementos y componentes correspondientes a los detalles internos del sistema.

La caja transparente determina los procesos y transformaciones que se requieren, es necesario considerar también los subprocesos, los cuales en combinación lograrían que el dispositivo produzca los resultados deseados.

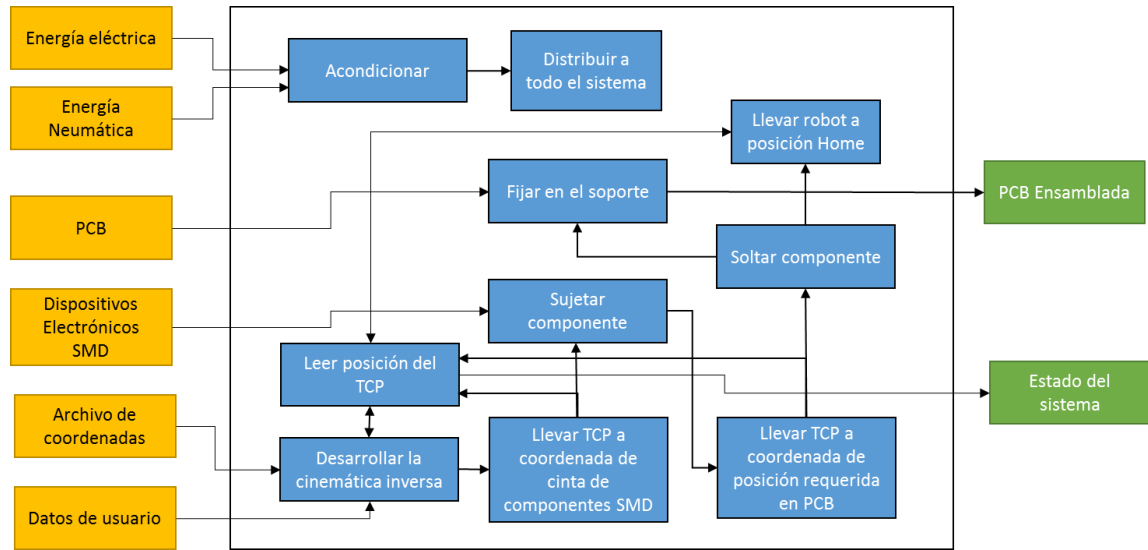
Los siguientes son detalles del proceso:

- La energía eléctrica es una de las principales entradas esta debe ser adecuada de tal forma que sea capaz de energizar cada parte del hardware, transformando la electricidad disponible.
- La energía neumática de ser necesaria debe regularse en cuanto a presión, y estar disponible para su uso, cuando sea necesario por medio de una válvula.
- El producto a obtener debe ser analizado para determinar su posición espacial dependiente del tiempo teniendo en cuenta su velocidad de desplazamiento, con esta información, detectar la entrada del producto al área de

acción; siguiendo con el proceso se generan dos trayectorias, la primera para alcanzar al producto y la segunda para desplazarlo a su posición, durante este proceso, paralelamente deben ser traducidas estas coordenadas a variables equivalentes para los mecanismos del dispositivo, esto se logra transformando por medio de una señal de inicio debe ser procesada para llevar al sistema a sus condiciones iniciales o como es bien conocido en la automatización a “home”, se debe obtener la posición actual del actuador y generar la trayectoria a la posición inicial definida.

- Una señal de pausa/reanudar debe detener al proceso y mantenerlo en las condiciones, capturando el estado actual para posteriormente reanudarlo.
- Una señal de parada debe indicarle al proceso que debe detenerse tal como la señal de pausa, diferenciándose en que, en este caso, el operario debe verificar y solucionar la falla en caso de que exista antes de poder reanudar el proceso.
- El estado del sistema son aquellas variables que describen el estado del sistema, las cuales deben estar a disposición como salidas, para su uso externo en sistemas periféricos o a disponibilidad de la siguiente etapa de producción.
- En el proceso, las principales acciones son controladas por una unidad de procesamiento encargada de realizar los cálculos y manejar el sistema por medio de actuadores.

Figura 27. Descomposición mediante caja transparente.



Fuente: Elaboración propia.

7.8. DISEÑO CONCURRENTE.

Partiendo de la base obtenida con anterioridad, se procede a plantear las diferentes alternativas posibles, tales que cumplan con las especificaciones, necesidades, métricas y funciones establecidas.

A continuación, se desglosan diferentes alternativas en la estructura del robot que servirán como base fundamental para la escogencia de la mejor solución. Cabe destacar que el punto de partida es el diseño de un robot Delta debido a las razones argumentadas en la justificación, por tanto, los diferentes conceptos que surgen serán sub-ensamblaje sub-procesos o sub-sistemas.

7.8.1. MORFOLOGÍA. A continuación, se evalúan tres tipos de morfologías de robots, con el objetivo de validar la implementación de la configuración Delta. Las configuraciones dispuestas para el análisis son: Cartesiano, Delta y Scara. Los robots cartesianos tienen una característica favorable y es que, debido a su cinemática, su implementación se puede desarrollar mediante motores paso a paso, los cuales tienen la ventaja de que su control es fácil y no requieren retroalimentación. La cinemática de los robots Scara también es sencilla, y su implementación estructural puede resultar ser simple. Sin embargo, los robots de configuración Delta tienen la gran ventaja en las altas velocidades que pueden otorgar, pues como se hizo mención en la justificación, de acuerdo a la forma de conexión de los motores, pueden alcanzar desplazamientos significativos con movimientos angulares en los motores relativamente pequeños (menores a 360°), hecho que no ocurre en las configuraciones Scara y Cartesianos.

Los criterios a evaluar son:

Velocidad, Precisión, Volumen de trabajo, Robustez y Estructura.

El criterio *estructura* corresponde a la complejidad mecánica.

*Para las siguientes matrices de selección, se toma una escala de 1 a 5, siendo 3 el valor medio de referencia.

Tabla 12. Validación selección morfología Delta.

		Concepto					
		Cartesiano		Delta		Scara	
		Concepto 1		Concepto 2		Concepto 3	
Criterio de selección	Peso	Valor	Evaluación ponderada	Valor	Evaluación ponderada	Valor	Evaluación ponderada
Velocidad	25%	2	0,50	5	1,25	4	1,00
Precisión	25%	5	1,25	3	0,75	3	0,75
Volumen*	15%	4	0,60	2	0,30	3	0,45
Robustez	15%	4	0,60	2	0,30	3	0,45
Estructura*	20%	2	0,40	5	1,00	2	0,40
	Total	3,35		3,60		3,05	
	Lugar	2		1		3	
	Continuar	No		Desarrollar		No	

Como se había mencionado en la justificación, el proyecto se implementa mediante un robot de configuración delta debido a sus importantes características en velocidad, dato que se valida mediante el estudio de comparación previo. (Tabla 9).

7.8.2. MOTORES. Como se vio en el análisis QFD, la necesidad más importante está relacionada con la resolución, precisión y exactitud del robot. Estas necesidades están asociadas directamente a la buena selección de los motores, pues son los que finalmente definen las limitaciones en dichas necesidades.

Se evalúan las siguientes cuatro opciones:

- Motor DC
- Motor Paso a paso
- Servomotor

A continuación, el detalle específico:

➤ **Opción 1: Motor DC.**

Ventajas:

- Más precisión.
- Más velocidad.
- Movimiento más suave.
- No hay pérdida de pasos.
- Resonancias mecánicas se pueden evitar.
- Rendimiento mejor.

➤ Desventajas:

- Coste.
- Complejidad del control.
- Escobillas (mantenimiento, chispas, zona muerta).

➤ **Opción 2: Motor Paso a paso.**

Ventajas:

- No necesitan realimentación.
- Baratos.
- Control fácil.
- No necesitan escobillas

➤ Desventajas:

- Mala resolución.
- Discontinuidad de velocidad (a bajas Velocidades).
- Pérdida de pasos.
- Resonancias mecánicas no se pueden evitar.
- Corriente de manteamiento alta.
- Rendimiento menor.

➤ **Opción 4: Servomotor²⁷.**

Ventajas:

- Es sumamente poderoso para su tamaño (mucho torque).
- Potencia proporcional para cargas mecánicas.
- No consume mucha energía.
- Mayor precisión.

Desventajas:

- No es posible cambiar las características eléctricas del motor por tanto no se puede cambiar la velocidad del mismo.
- Está limitado por el circuito de control a únicamente variar de 0° a 180°.
- Su construcción puede ser costosa.
- Poder controlar varias secuencias de posiciones, es difícil sin utilizar sistemas basados con micros.

La selección del tipo de motor a usar dependerá entonces de las siguientes características:

Velocidad, Control, Fuerza, Precisión y Consumo de potencia.

²⁷ CANDELAS HERÍAS, Francisco A. & Juan A. Corrales Ramón, [En línea] Servomotores, 2007. [Consultado abril 2016] Disponible en internet: <http://www.aurova.ua.es/previo/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>

Tabla 13. Selección de motores.

Motores							
		Motor DC		Motor PAP		Servomotor	
		Concepto 1		Concepto 2		Concepto 3	
Criterio de selección	Peso	Valor	Evaluación ponderada	Valor	Evaluación ponderada	Valor	Evaluación ponderada
Velocidad	25%	5	2,25	5	1,25	3	1,25
Control	25%	3	1,25	5	2,25	5	2,25
Fuerza	15%	1	0,15	3	0,75	3	0,75
Precisión	25%	1	0,25	3	1,25	5	2,25
Consumo	10%	3	0,50	1	0,10	3	0,50
	Total	2,70		3,80		4,00	
	Lugar	3		2		1	
	Continuar	No		No		Desarrollar	

De acuerdo a la tabla 9, en donde se realiza una comparación entre las tres opciones de motores, se determina que el tipo de motor a usar será Servomotor.

7.8.3. Controlador. El sistema electrónico debe ser capaz de responder de manera oportuna a las órdenes provenientes del software de control, datos correspondientes a la generación de las trayectorias mediante la cinemática inversa del robot. Es importante tener en cuenta que el controlador deberá responder de manera rápida, por lo cual, la velocidad de procesamiento debe ser elevada, al igual que la velocidad de comunicación serial hacia el computador.

Se designan cuatro opciones que corresponden a:

- Arduino
- Microcontrolador PIC
- OpenCM

➤ Opción 1: Arduino.

Ventajas²⁸

- Arduino simplifica el proceso de trabajar con microcontroladores.
- Las placas Arduino son más accesibles comparadas con otras plataformas de microcontroladores. Los módulos más caros de Arduino pueden ser montadas a mano bajando sus costos.
- El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux; mientras que la mayoría de otros entornos para microcontroladores están únicamente limitados a Windows.
- El entorno de programación de Arduino es fácil de usar para principiantes y flexible para los usuarios avanzados. Además, Arduino está basado en el entorno de programación de Processing, con lo que los estudiantes que aprendan a programar en este entorno se sentirán familiarizados con el entorno de desarrollo Arduino.
- El software Arduino es de distribución de licencia libre y preparado para ser adaptado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y en caso de querer profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en los programas de los usuarios, si es que así lo desean.
- Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

Desventajas

- Dado que la programación no se realiza en ensamblador, el precio a pagar por el uso de las librerías es un retraso en la ejecución de las instrucciones, algunos microsegundos que en el caso de dispositivos de uso cotidiano son irrelevantes, pero significativos a la hora de hacer adquisición de datos.

²⁸ Arduino, [En línea] Arduino, 2016. [Consultado marzo 2016] Disponible en internet: <https://www.arduino.cc/>

- El hecho de que la plataforma venga ya ensamblada le quita flexibilidad a los proyectos, así por ejemplo se obliga a usar un espacio y forma acorde con el PCB del arduino, para superar esto, se debe trabajar con un microcontrolador diferente al de la plataforma y diseñar las PCB desde cero como con los PICs.

➤ Opción 2: Microcontrolador PIC.

Ventajas²⁹

- Los microcontroladores PIC son fácil de grabar, ya que solo necesitamos una computadora o una notebook. Un PIC se puede grabar mediante diferentes puertos, como por ejemplo el serie o el USB. Estos son los dos más populares que utilizan las plaquetas grabadoras convencionales.
- La gran ventaja de un PIC que no existía antes, es que nos permite controlar, programar y sincronizar tareas electrónicas a través del tiempo simplemente realizando una correcta programación. En el pasado no muy lejano, esto no era posible ya que para controlar cada proceso era necesario un circuito muy complicado y específico para cada cosa que se necesite. En cambio, con un microcontrolador PIC, este circuito integrado hace todo por nosotros.
- En el mercado existen varios softwares que nos ayudan a programar un microcontrolador de este tipo, como por ejemplo el PICC, o el MPLAB, es decir, que los PIC, están muy extendidos y difundidos en la electrónica actual.
- Existe una gran diversidad de microcontroladores PIC en el mercado de Microchip y ésta también es una gran ventaja, ya que podemos elegir entre diversas características que uno no tiene pero otro sí, como cantidad de puertos, cantidad de entradas y salidas, conversor Analógico a Digital, cantidad de memoria, espacio físico, y este tipo de cualidades que nos permiten tener una mejor elección de un PIC.

Desventajas

- Una de las desventajas de un microcontrolador PIC es que se necesitan llamar a muchas instrucciones para realizar una tarea en particular. Esto siempre y cuando el proyecto sea complejo.

²⁹ GONZÁLEZ V, José Adolfo. Introducción a los microcontroladores, McGraw Hill, 1994.p15

- Otra de las desventajas pero no tan significativas o importantes es que los PIC no son tan baratos como uno los puede esperar. Comprar un PIC puede no ser tan caro pero tampoco barato.

➤ **Opción 3: OpenCm.**

Ventajas

- Programación similar al Arduino.
- Es un producto económico. El costo es similar al precio de un Arduino.
- La velocidad de comunicación puede alcanzar los 4Mbps.
- Posee una arquitectura de 32bits.
- Es robusto.
- Existe mucha documentación para hacer control con motores de la misma marca, por lo cual el control resulta ser más sencillo.
- Posee puertos especiales para la conexión de motores de la misma marca.
- Tiene pines de libre configuración entre los cuales tiene salidas a PWM, comunicación I²C y convertidor ADC.

Desventajas

- Es un producto exclusivo para el control de motores de la misma marca (Robotis).
- No tiene case ni protección al ruido electromagnético.

Teniendo en cuenta las ventajas y desventajas para cada opción, se establecen cinco criterios a través de los cuales se realizará la evaluación de la selección del tipo de controlador a usar.

Criterios: Costo, Robusto, Arquitectura, Velocidad, Consumo.

Tabla 14. Selección del Controlador.

Selección del Controlador							
		Arduino		OpenCM		Microcontrolador PIC	
		Concepto 1		Concepto 2		Concepto 3	
Criterio de selección	Peso	Valor	Evaluación ponderada	Valor	Evaluación ponderada	Valor	Evaluación ponderada
Costo	20%	3	1,00	1	0,20	5	1,80
Robusto	25%	3	1,25	5	2,25	1	0,25
Arquitectura	15%	3	0,75	5	1,35	1	0,15
Velocidad	25%	3	1,25	3	1,25	3	1,25
Consumo	15%	3	0,75	1	0,15	5	1,35
	Total	3,00		3,10		2,90	
	Lugar	2		1		3	
	Continuar	No		Desarrollar		No	

La tabla 10 arroja que el controlador a implementar será el correspondiente al concepto 2: OpenCM.

7.8.4. SOFTWARE. El software a desarrollar debe diseñarse en un lenguaje de programación que sea práctico, robusto, orientado a objetos y en lo posible, open software.

Las opciones a evaluar son:

- C#.
- Java.
- Python.

➤ **Opción 1: C#.**

Ventajas:

Las principales ventajas que presenta el uso C# en comparación con otros lenguajes es su potencia como lenguaje, pero también su flexibilidad. Soporta la

mayoría de paradigmas, destacando el paradigma funcional que combinado con el paradigma orientado a objetos hacen del lenguaje uno de los más potentes.

Desventajas

La principal desventaja que puede presentar respecto a Java radica en su portabilidad y en la dificultad que trae un desarrollo completo de un producto software empleando este lenguaje. Java es un lenguaje que presenta una curva de aprendizaje menor que la de lenguajes como C# o Perl. Esta curva de aprendizaje no se debe a la sintaxis del código, ni si quiere a como se programa o los nuevos conceptos que introduce, es una curva más referente a la dificultad que con lleva para un usuario novel realizar una correcta documentación del código o una ausencia de una IDE que sirva como referente en el sector educativo, en Java si tenemos el famoso BlueJ utilizado en tantas universidades.³⁰

➤ Opción 2: Java.

Ventajas

- Manejo automático de la memoria. El manejo de la memoria se hace automáticamente y utilizando el garbage collector. Esto ahorra *significante* tiempo de programación.
- Lenguaje Multi-plataforma: El código que escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma.
- Programación Orientada a Objetos: Paradigma muy utilizado hoy en día que facilita y organiza mucho la programación.
- Puede correr en el explorador y en dispositivos móviles.
- Sintaxis similar a C/C++, pero más simple.
- Fácil de aprender.
- Gratuito

³⁰ Desarrollo de software con C#, [En línea] ROOTEAR, 2015. [Consultado marzo 2016] Disponible en internet: <http://rootear.com/desarrollo/desarrollo-software-csharp>.

Desventajas

- Menos Eficiente, comparado a C/C++.
- Requiere un intérprete.
- Algunas implementaciones y librerías pueden tener código rebuscado.
- Una mala implementación de un programa en java, puede resultar en algo muy lento.
- Algunas herramientas tienen un costo adicional.

➤ Opción 3: Python.

Ventajas

- Rápido de desarrollar.
- Sencillez y velocidad.
- Sus bibliotecas hacen gran parte del trabajo.
- Soporta varias bases de datos.

Desventajas:

- Los programas interpretados son más lentos que los compilados.

Debido a su alto desempeño, potencia e importantes características, se evalúan estos tres conceptos con respecto a los criterios más sobresalientes que se extraen de las ventajas y desventajas mencionadas con anterioridad.

Criterios: Flexible, Multiplataforma, Multiparadigma, Eficiente, Costo Licencia, Sencillo.

Tabla 15. Selección de plataforma de programación.

Selección de plataforma de programación							
		C#		Java		Python	
		Concepto 1		Concepto 2		Concepto 3	
Criterio de selección	Peso	Valor	Evaluación ponderada	Valor	Evaluación ponderada	Valor	Evaluación ponderada
Flexible	10%	5	0,90	3	0,50	3	0,50
Multiplataforma	30%	5	2,70	5	2,70	5	2,70
Multiparadigma	15%	5	1,35	5	1,35	5	1,35
Eficiente	15%	3	0,75	5	1,35	3	0,75
Costo Licencia	20%	1	0,20	5	1,80	5	1,80
Sencillo	10%	1	0,10	3	0,50	3	0,50
	Total	3,50		4,60		4,30	
	Lugar	3		1		2	
	Continuar	No		Desarrollar		No	

De acuerdo a la matriz anterior, el lenguaje de programación a utilizar será **JAVA**.

7.8.5. Mecanismo. El mecanismo del robot Delta se evalúa en las dos opciones clásicas de configuración que corresponden a sistema con actuadores lineales y sistema con actuadores rotacionales. El detalle a continuación.

➤ **Opción 1: Actuadores lineales³¹.**

Ventajas

- Movimientos suaves.
- Mayor volumen de trabajo.
- Operación en eje Z limitado por la longitud de los actuadores.
- Cinemática más sencilla respecto del robot con actuadores rotacionales

³¹ R.L. Williams II. The Delta Parallel Robot: Kinematics Solutions[En línea] ohio [Consultado enero 2016]. Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf,

Desventajas

- Requiere correas dentadas para la transmisión del movimiento.
- Estructura grande, sobre todo en altura.

Figura 28. Robot delta con actuadores lineales.



Fuente: R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, 2016. [En línea]. ohio. [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

➤ Opción 2: Actuadores rotacionales.

Ventajas

- Ocupa menos espacio que el robot con actuadores lineales.
- Puede ser instalado sobre pedestales en su zona superior.
- Mucho más veloz que el anterior.
- Mayor facilidad en el mantenimiento.

Desventajas

- Requiere motores de mayor resolución.
- Cinemática compleja.
- En el control, se debe tener en cuenta la gravedad cuando los actuadores están alineados horizontalmente.

Figura 29. Robot Delta con actuadores rotacionales.



Fuente: FANUC, M-3iA Delta Robot, 2014. [En línea]. Robot.fanucamerica [Consultado Enero 2016] Disponible en internet: http://robot.fanucamerica.com/robotics-articles/delta_robots_improve_highly_repetitive_tasks.aspx

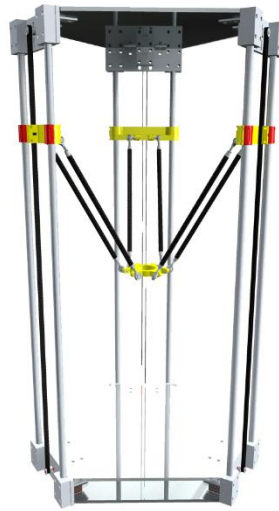
➤ Conceptos

Concepto 1.

Este concepto tiene la enorme ventaja de que el movimiento es suave e independientemente de la resolución de los motores, es posible aumentar la precisión añadiendo sistemas de reducción mecánica mediante poleas de manera fácil. Este concepto combina el uso de actuadores lineales mediante el uso de motores paso a paso debido a su facilidad de control, bajo costo, y alta precisión en lazo abierto.

El controlador a usar es un Arduino debido a que en la red ya existe documentación para control de motores paso a paso, destinado a la implementación en robots Delta.

Figura 30. Concepto 1, mecanismo con actuadores lineales.



Fuente: SHADOW FAX MACHINE, THE DELTA ROBOT, 2016. [En línea]. shadowfaxmachine [Consultado mayo 2016] Disponible en internet: <http://shadowfaxmachine.squarespace.com/>

➤ **Concepto 2.**

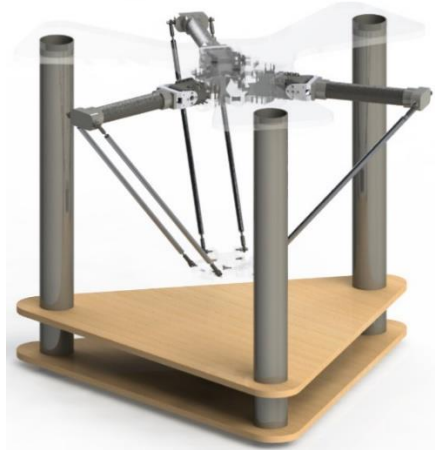
Este concepto se basa en la implementación estructural mediante actuadores rotacionales, los cuales permiten obtener un área de trabajo más cómoda. Ahora, teniendo en cuenta que la aplicación estará destinada al posicionamiento de componentes electrónicos SMD, no se requiere una estructura de altura significativa, por lo cual el volumen de trabajo con respecto a la altura no es una limitante.

Los motores a usar son Servos debido a que internamente tienen ya todo un sistema mecánico para reducción de velocidad. Por otro lado, poseen sistemas electrónicos embebidos, de tal modo que el control resulta ser más sencillo. La siguiente Figura ilustra el concepto, donde se observa que los tres motores están acoplados directamente a los brazos.

El controlador a usar es el OpenCM debido a que estos son totalmente compatibles con servomotores y la comunicación a ellos se realiza de manera serial, lo que facilita el control.

Este concepto posee una base móvil con el objetivo de poder desplazar la PCB a diferentes alturas, dependiendo del área de la misma, de tal manera que podrá procesar tarjetas de mayor tamaño, esto gracias a la posibilidad de variar el centro del volumen de trabajo.

Figura 31. Concepto 2, mecanismo con actuadores rotacionales y base móvil.



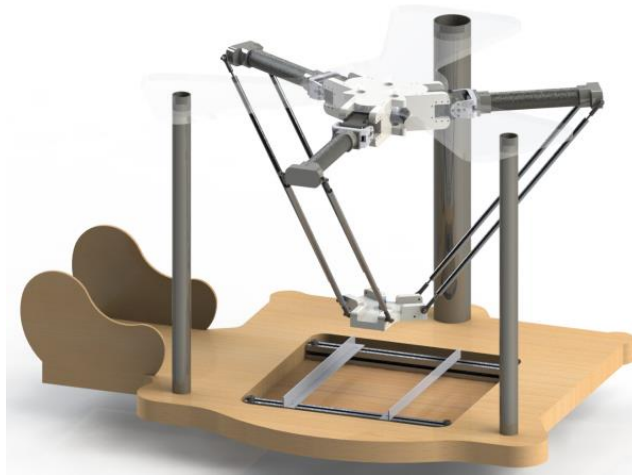
Fuente: Elaboración propia.

➤ **Concepto 3.**

El presente concepto es similar al anterior, a diferencia de que éste implementa una base estática que permita fijar la PCB a la estructura, brindando mayor robustez en el momento de operación. En la zona izquierda de la siguiente Figura se observa la zona que corresponde a la ubicación de la porta cintas de componentes.

Éste concepto brinda mayor seguridad al operador y mayor confianza durante el posicionamiento.

Figura 32. Concepto 3, mecanismo con actuadores rotacionales y base fija.



Fuente: Elaboración propia.

7.9. SELECCIÓN DE CONCEPTOS.

Finalmente, se procede a evaluar los conceptos de acuerdo a los criterios de selección relacionados con las necesidades. La siguiente matriz pondera cada concepto con el respectivo criterio, obteniendo así un nivel cuantitativo con el cual se escogerá el concepto definitivo. El concepto a desarrollar será el número 3.³²

Tabla 16. Evaluación de los conceptos

Concepto							
		Actuador lineal		Actuador Rotacional		Actuador Rotacional	
		Concepto 1		Concepto 2		Concepto 3	
Criterio de selección	Peso	Valor	Evaluación ponderada	Valor	Evaluación ponderada	Valor	Evaluación ponderada
Preciso	20%	9	1,80	9	1,80	5	1,00
Exacto	20%	5	1,00	5	1,00	5	1,00
Alta resolución	20%	5	1,00	5	1,00	5	1,00
Robusto	15%	9	1,35	1	0,15	9	1,35

³² ULRICH, Karl T., EPPINGER, Steven D. Op. cit., p. 154.

Seguro	5%	5	0,25	1	0,05	9	0,45
Veloz	20%	1	0,20	5	1,00	5	1,00
	Total	5,60		5,00		5,80	
	Lugar	2		3		1	
	Continuar	No		No		Desarrollar	

7.9.1. Efecto final. Debido a que los componentes electrónicos de montaje superficial tienen dimensiones de tamaños milimétricos, el efecto final debe permitir sujetarlos de tal manera que se garantice un eficiente agarre, evitando una manipulación tal que los componentes sean maltratados. Para hacer una aproximación al tamaño, se toma como ejemplo los elementos de montaje superficial tipo 1206 cuyas dimensiones son de 3.2mm x 1.6mm.

Un aspecto importante a tener en cuenta es que se debe evitar que haya algún tipo de inducción electromagnética o contacto eléctrico, pues esto podría afectar directamente el deterioro de los elementos, por tanto, el efecto debe ser un elemento en lo posible libre de conexiones eléctricas o terminaciones metálicas.

Teniendo en cuenta lo anterior, resulta eficiente utilizar una ventosa para el agarre de los componentes, pues el suministro activo es el aire comprimido, elemento que no afecta los dispositivos al contacto.

7.10. ARQUITECTURA DEL PRODUCTO.

El producto se ha segmentado en siete partes, donde se resaltan los más importantes sub-sistemas, donde se tiene:

Sistema Informático: Aplicación (Software) encargado de procesar la información ECAD de la PCB y generar las trayectorias sobre el robot.

Sistema Electrónico: Sistema embebido encargado de procesar los datos de posicionamiento de los motores para conseguir una traslación X Y Z en el TCP.

Sistema de Posicionamiento: Estructura funcional compuesta por los motores correspondientes a los brazos articulados.

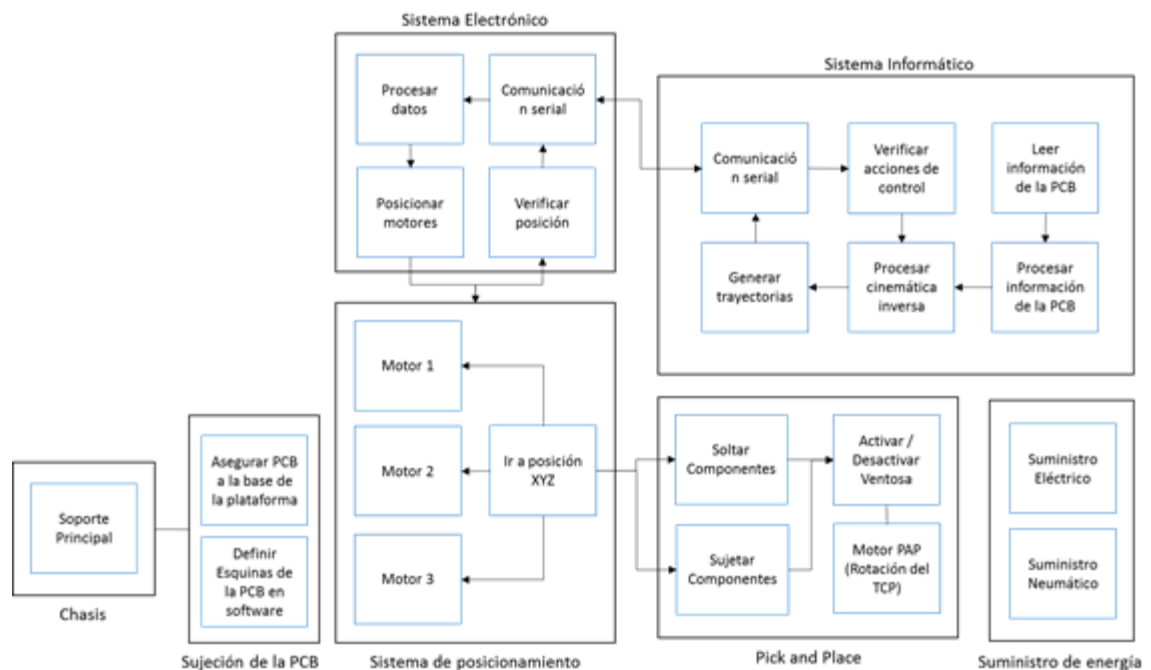
Pick and Place: Efecto final o TCP (Sistema de sujeción y soltado por ventosa).

Chasis: Soporte estructural base.

Sujeción de la PCB: Sistema de fijación de la tarjeta al chasis del robot.

Suministro de energía: Suministro de energía eléctrica y neumática.

Figura 33. Arquitectura del robot Delta.

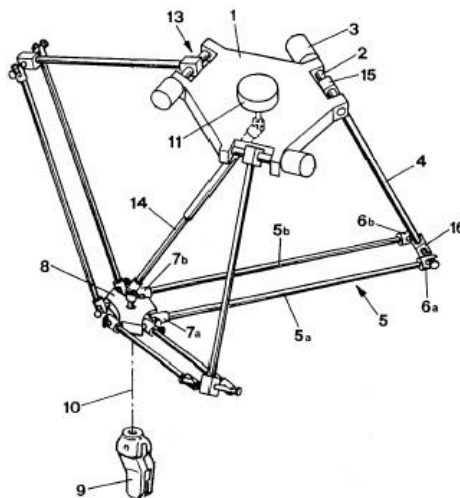


Fuente: Elaboración propia.

8. DISEÑO DETALLADO

Se desarrollará un robot Delta de cuatro grados de libertad, 3 para conseguir la traslación X , Y , Z del efector, y un cuarto grado de libertad para conseguir la rotación del efector final. A continuación, se ilustra en la Figura 34, un bosquejo de lo que representaría el diseño base del robot.

Figura 34. Diseño de robot delta.



Fuente: R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, 2016. [En línea]. ohio [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

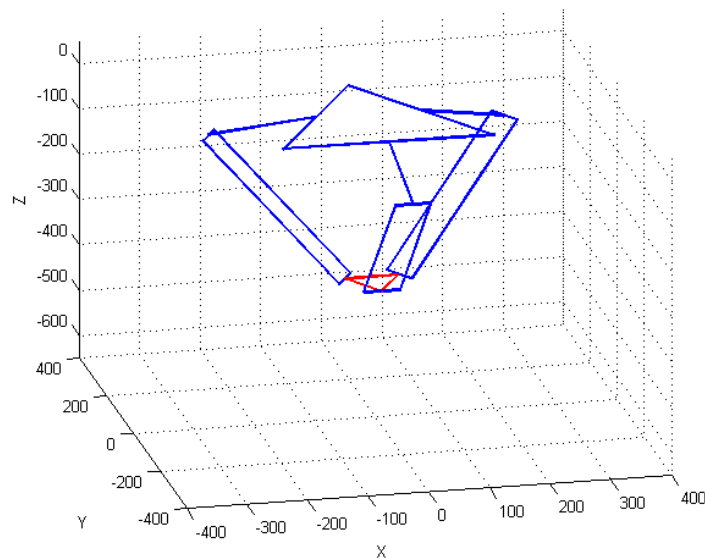
El diseño del robot viene ligado al resultado de su cinemática inversa, pues las longitudes de cada articulación afectarán directamente la posición final del efector, así como la precisión. Por ejemplo, entre más largo son los brazos articulados, se pierde en resolución, por lo cual la precisión se ve afectada. Por tanto, se ha modelado en MatLab diferentes configuraciones, bajo diversas longitudes con el objetivo de obtener un balance entre la resolución que se necesita y la resolución que se puede brindar, es decir, la combinación de dichas variables con el volumen de trabajo a establecer y la resolución de los motores definirán la resolución y precisión final del robot.

8.1. DISEÑO DE SISTEMA MECÁNICO

Como se había mencionado, se parte del modelo base del robot, en donde se observa en la Figura 35 la presencia de los elementos fundamentales del robot delta:

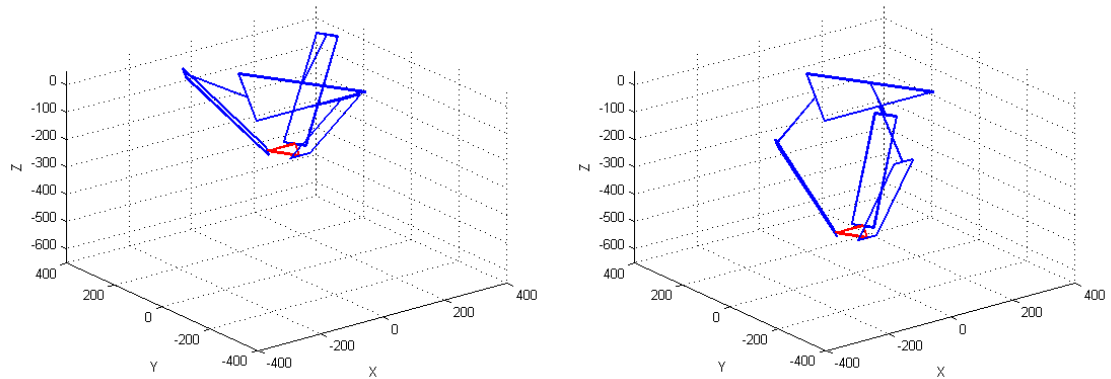
La base superior en forma triangular, donde los puntos medios de cada lado son los puntos de conexión con los brazos superiores. En dichos puntos se conectan los motores, indicando así que el movimiento se realiza de manera angular. Se podría deducir en este punto que, si los tres motores se mueven de manera sincronizada en el mismo sentido, el efector final sube o baja de manera lineal (ver Figura 36).

Figura 35. Diseño base del robot.



Fuente: Elaboración propia.

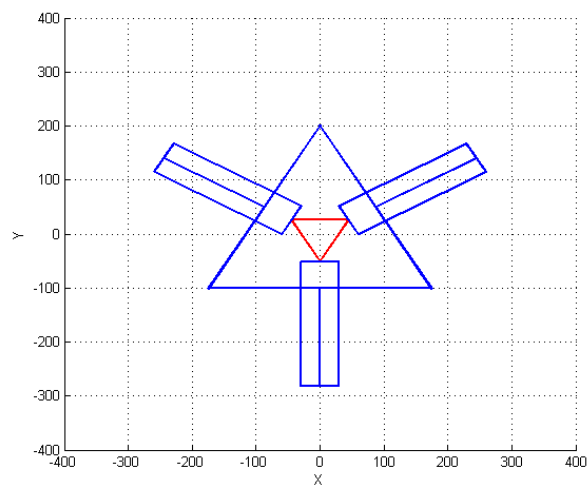
Figura 36. Movimiento sincronizado de motores.



Fuente: Elaboración propia.

La Figura 37 ilustra una vista superior del diseño base, en donde se observa que cada brazo articulado es conectado a cada lado del triángulo que corresponde a la base superior, consiguiendo de esta manera, tres cadenas cinemáticas que serán evaluadas posteriormente.

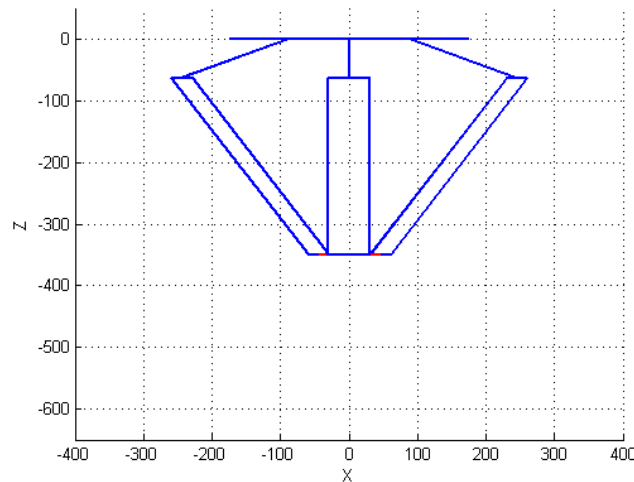
Figura 37. Vista en planta del diseño base.



Fuente: Elaboración propia.

En la Figura 38 se observa que el sistema de referencia en Z se encuentra en la base superior, por tanto, todas las coordenadas Z del espacio de trabajo serán negativas.

Figura 38. Vista frontal del diseño base.



Fuente: Elaboración propia.

En la Figura 38 se observa también la conexión de los brazos superiores a los brazos inferiores, los cuales forman tres paralelogramos que tienen como objetivo el mantener siempre horizontal el efector, consiguiendo así únicamente movimientos de traslación, guardando siempre los tres grados de libertad en relación a los tres grados de movilidad (motores). Para esto, los brazos superiores deben conectarse a los brazos inferiores mediante articulaciones de tipo rotacional (ver Figura 39) en los 12 puntos de conexión (6 arriba, 6 abajo: 4 para cada brazo). En este orden de ideas, los brazos inferiores se conectan al efector final o TCP de la misma manera.

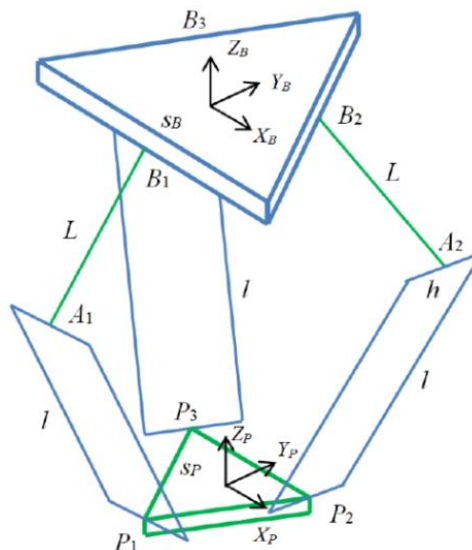
Figura 39. Articulación rotacional (1 GDL).



Fuente: ZOTOVIC Ranko, Principios básicos de la robótica, [Diapositivas], Valencia: Universidad Politécnica de Valencia, 2014. Diapositiva 36.

8.1.1. Modelo general del robot. De este modo, partiendo de la fundamentación realizada, se procede a modelar cada pieza del robot, obteniendo así las ecuaciones que serán base fundamental para el desarrollo cinemático. Para empezar, se detallan las variables generales como lo ilustra la Figura 40.

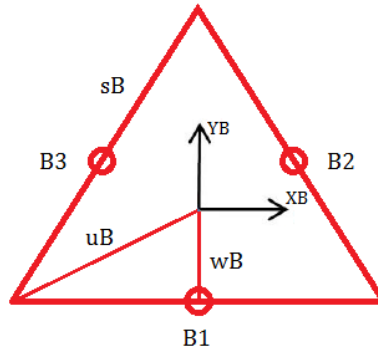
Figura 40. Diagrama general del robot.



Fuente: R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, 2016. [En línea]. ohio [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

Ahora bien, antes de definir las ecuaciones generales, se detallan los aspectos relevantes de la base y el TCP como lo ilustran las Figuras 36 y 37.

Figura 41. Detalles de la base del robot.



Fuente: Elaboración propia.

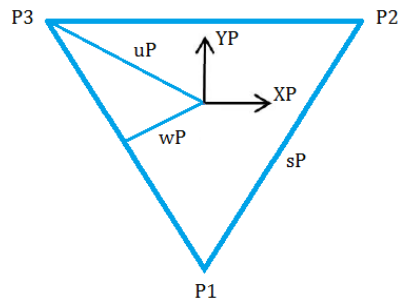
En la Figura 41 se observa el detalle de la base, siendo:

- **B1, B2 y B3** los puntos donde se articulan los brazos superiores mediante los motores.
- **sB** la longitud del lado de la base.
- **uB y wB** las distancias del centro de referencia de la base a sus extremos.

En la Figura 42 se observa el detalle del TCP, siendo:

- **P1, P2 y P3** los puntos donde se articulan los brazos inferiores mediante los paralelogramos.
- **sP** la longitud del lado del TCP
- **uP y wP** las distancias del centro de referencia del TCP a sus extremos

Figura 42. Detalle del TCP del robot.



Fuente: Elaboración propia.

A continuación, se denotan los vectores con referencia a la Base y al TCP.

Vectores con respecto al sistema de referencia de la base B:

$${}^B B_1 = \begin{Bmatrix} 0 \\ -wB \\ 0 \end{Bmatrix} \quad {}^B B_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2} wB \\ \frac{1}{2} wB \\ 0 \end{Bmatrix} \quad {}^B B_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2} wB \\ \frac{1}{2} wB \\ 0 \end{Bmatrix}$$

Vectores con respecto al sistema de referencia del TCP P:

$${}^P P_1 = \begin{Bmatrix} 0 \\ -uP \\ 0 \end{Bmatrix} \quad {}^P P_2 = \begin{Bmatrix} \frac{sP}{2} \\ wP \\ 0 \end{Bmatrix} \quad {}^P P_3 = \begin{Bmatrix} -\frac{sP}{2} \\ wP \\ 0 \end{Bmatrix}$$

Vectores con respecto a los vertices de la base:

$${}^B b_1 = \begin{Bmatrix} \frac{sB}{2} \\ -wB \\ 0 \end{Bmatrix} \quad {}^B b_2 = \begin{Bmatrix} 0 \\ uB \\ 0 \end{Bmatrix} \quad {}^B b_3 = \begin{Bmatrix} -\frac{sB}{2} \\ -wB \\ 0 \end{Bmatrix}$$

Dónde:

$$wB = \frac{\sqrt{3}}{6} sB \quad uB = \frac{\sqrt{3}}{3} sB \quad wP = \frac{\sqrt{3}}{6} sP \quad uP = \frac{\sqrt{3}}{3} sP$$

Para calcular la movilidad del robot y determinar el número de grados de libertad, se recurre a la ecuación de movilidad espacial de Kutzbach-Gruebler.³³

$$M = 6(N - 1) - 5J_1 - 4J_2 - 3J_3$$

Siendo:

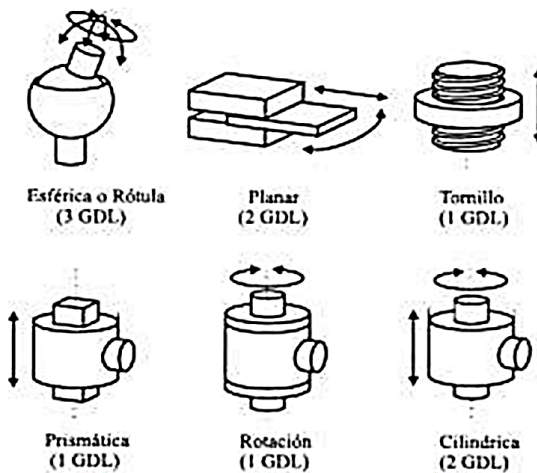
- M Movilidad (Número de grados de libertad)

³³, Conceptos y definiciones en cinemática, [En línea] Universidad de Huelva 2008. [Consultado marzo 2016] Disponible en internet: <http://www.uhu.es/rafael.sanchez/ingenieriamaquinas/carpetaapuntes.htm/Apuntes%20Tema%201.pdf>.

- N Número total de eslabones
- J1 Número de articulaciones de 1 grado de libertad
- J2 Número de articulaciones de 2 grados de libertad
- J3 Número de articulaciones de 3 grados de libertad

A continuación, diferentes tipos de articulación de acuerdo al número de grados de libertad.

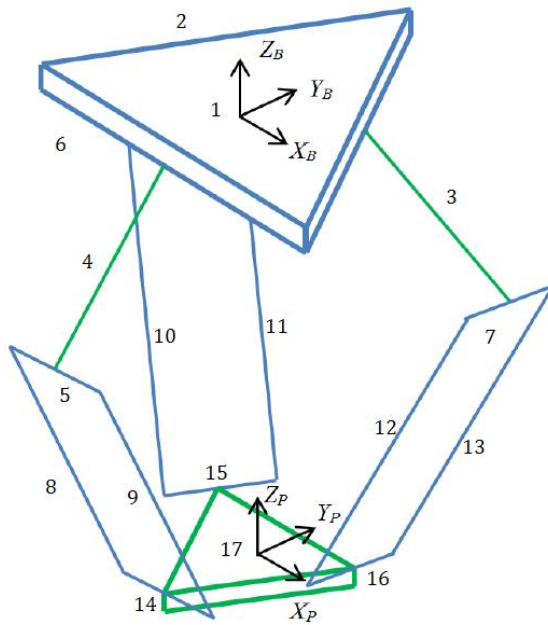
Figura 43. Tipos de articulaciones básicas.



Fuente: ZOTOVIC Ranko, Principios básicos de la robótica, [Diapositivas], Valencia: Universidad Politécnica de Valencia, 2014. Diapositiva 36.

Para determinar la movilidad, se debe obtener el número de eslabones y el número de articulaciones.

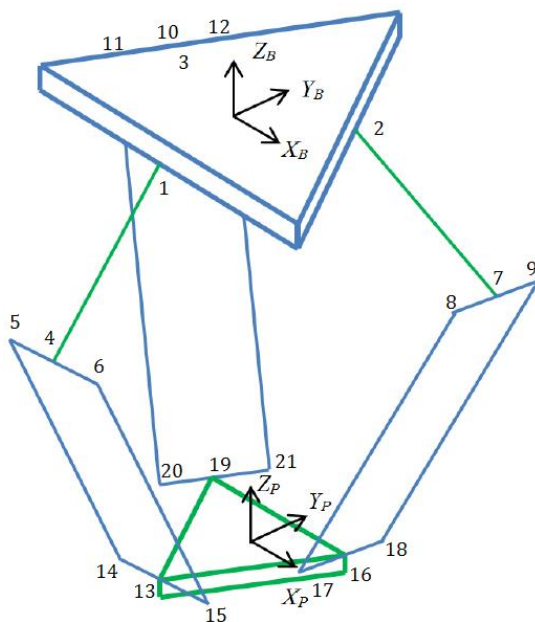
Figura 44. Eslabones del robot.



Fuente: Elaboración propia.

- Los eslabones 1 y 17 corresponden a la base y el TCP respectivamente.
- Los eslabones 2, 3 y 4 corresponden a los brazos superiores.
- Los eslabones 5, 6 y 7 corresponden a las barras horizontales que conectan con los brazos superiores.
- Los eslabones 8, 9, 10, 11, 12 y 13 corresponden a las barras paralelas de los paralelogramos.
- Los eslabones 14, 15 y 16 corresponden a las barras horizontales que conectan con el TCP.

Figura 45. Articulaciones del robot.



Fuente: Elaboración propia.

- Los puntos 1, 2 y 3 corresponden a las uniones entre base y brazos superiores.
- Los puntos 4, 5, 6, 7, 8, 9, 10, 11, y 12 corresponden a las uniones entre los brazos superiores y los brazos inferiores (paralelogramos).
- Los puntos 13, 14, 15, 16, 17, 18, 19, 20 y 21 corresponden a las uniones entre los brazos inferiores (paralelogramos) y el efector final o TCP.

De acuerdo a los datos obtenidos de eslabones y articulaciones, ilustrados en las Figuras 39 y 40, se calcula la movilidad:

Para el presente diseño:

- N 17
- J1 21
- J2 0
- J3 0

$$M = 6(17 - 1) - 5(21) - 4(0) - 3(0)$$

$$M = -9 \text{ Grados de Libertad}$$

Como se había mencionado, la movilidad debería ser 3GDL, sin embargo, el resultado obtenido indica que el robot delta es una estructura severamente sobre-restringida y estáticamente indeterminada, pero esto es incorrecto. Ahora, debido a que se tienen tres mecanismos de cuatro barras paralelas, al quitar una barra en cada mecanismo, la cinemática del robot funcionaría de manera idéntica.

Por tanto, el número de eslabones se reduciría en 3 y el número de articulaciones en se reduciría en 6, obteniendo:

- N 14
- J1 15
- J2 0
- J3 0

$$M = 6(14 - 1) - 5(15) - 4(0) - 3(0)$$

$$M = 3 \text{ Grados de Libertad}$$

8.1.2. Dimensionamiento y espacio de trabajo.

Las dimensiones escogidas para el diseño estructural del robot son:

Tabla 17. Dimensiones del robot.

Nombre	Descripción	Valor (mm)
sB	Lado de la base	350
uB	Longitud del centro de la base a el vértice	202
wB	Longitud del centro de la base a los motores	101
sP	Lado del TCP	90
uP	Longitud del centro del TCP al punto de articulación	52
wP	Longitud del centro del TCP al punto medio del lado	26
L	Longitud de brazos superiores	192
I	Longitud de brazos inferiores	368
A	Ancho de los mecanismos de cuatro barras	60

Fuente: Elaboración propia.

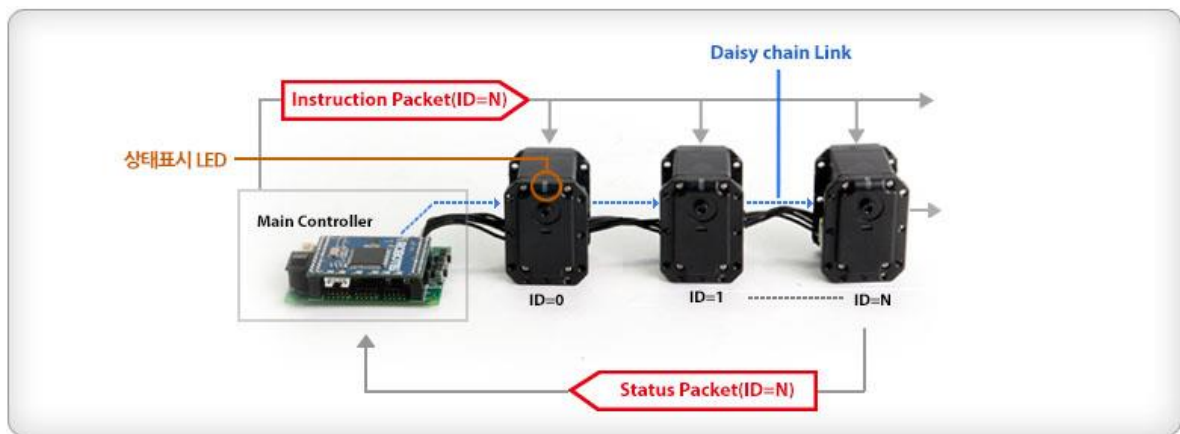
8.1.3. Selección de motores. Uno de los aspectos más importantes que determinan la resolución y precisión de un robot Delta, está en la calidad y tipo de motores utilizados, esto debido a que, para conseguir una posición dada dentro del volumen de trabajo establecido, los motores ejecutan trayectorias angulares que, para el presente proyecto, no superan los 360 grados, es decir, un minúsculo cambio en el ángulo del eje, produce una traslación significativa en el efector final. -La significancia en la traslación estará definida por las longitudes de los brazos articulados-. Por dichas razones, la selección de los motores es un proceso vital, en el que se debe ser muy cuidadoso, sobre todo con respecto a las características de resolución del mismo. Teniendo en cuenta la matriz de tamizaje en Tabla 9, se descartaron los menos convenientes y se escoge el motor de tipo Servo los cuales entre sus diferentes características, tienen incorporado todo un sistema de engranajes para reducción de velocidad, un *encoder* para el control de la posición y un sistema electrónico.

Para mayor practicidad en el desarrollo del sistema electrónico y el sistema de control, se opta por usar servo motores tipo *Smart*, los cuales tienen incorporado, además, todo un completo sistema de comunicación y un avanzado sistema de

control, no sólo de posición, sino también de fuerza y velocidad, con la posibilidad de medir en tiempo real variables físicas como la temperatura, el voltaje de operación, la corriente de armadura, entre otras.

La marca escogida es Robotis, quienes tienen en su línea de productos el motor Dynamixel, distinguido porque se controla mediante comunicación serial asincrónica; poseen en su interior un microcontrolador de 32 bits y permiten crear una red tipo bus (ver Figura 46), acotada por la capacidad de la fuente de alimentación.

Figura 46. Ejemplo de conexión de una red de motores Dynamixel.



Fuente: ROBOTIS, Status Display functions, 2016. [En línea]. Robotis, [Consultado Marzo 2016] Disponible en internet: http://en.robotis.com/index/product.php?cate_code=101010

Como el torque que deben soportar los brazos superiores no supera 1Nm, se opta por usar el motor:

Dynamixel AX-12A.

Motor cuyas características se ilustran en la tabla 15.

Tabla 18. Características del motor seleccionado.

Peso	54.6g
Dimensiones	32mm * 50mm * 40mm
Resolución	0.29°
Relación de reducción	254:01:00
Torque	1.5N.m (a 12.0V, 1.5A)
Velocidad sin carga	59rpm (a 12V)
Rango de posición angular	0° ~ 300° Continuo
Temperatura de operación	-5°C ~ +70°C
Voltaje	9 ~ 12V (Voltaje recomendado: 11.1V)
Señal de mando	Paquetes digitales
Tipo de protocolo	Comunicación serie asincrónica <i>Half duplex (8bit, 1stop, No Parity)</i>
Enlace	TTL
ID	254 ID (0~253)
Velocidad de Comunicación	7343bps ~ 1 Mbps
Realimentación	Posición, Temperatura, Carga, Voltaje de entrada, etc.
Material	Plástico de ingeniería

Fuente: ROBOTIS, AX-12/ AX-12+/ AX-12A, 2016. [En línea] ROBOTIS,. [Consultado Marzo 2016] Disponible en internet: http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm

Figura 47. Motor seleccionado.



Fuente: ROBOTIS, AX-12/ AX-12+/ AX-12A, 2016. [En línea]. ROBOTIS, [Consultado Marzo 2016] Disponible en internet: http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm

El motor escogido con antelación se replica 3 veces y corresponde los elementos que sujetarán los tres brazos del robot. Pero hace falta un último motor, el cual corresponde al que se debe acoplar a la vetosa para lograr la rotación de la misma. Este no es crítico, pues la rotación deseada de los componentes electrónicos dentro de la PCB está limitada a giros de 90 grados. Por tanto, se escogen motores paso a paso de 15° de resolución y voltaje de operación 12V, debido a que no requieren realimentación, son precisos y su control es ligeramente sencillo. La tabla 4 ilustra las características eléctricas y mecánicas.

Figura 48. Motor paso a paso unipolar.

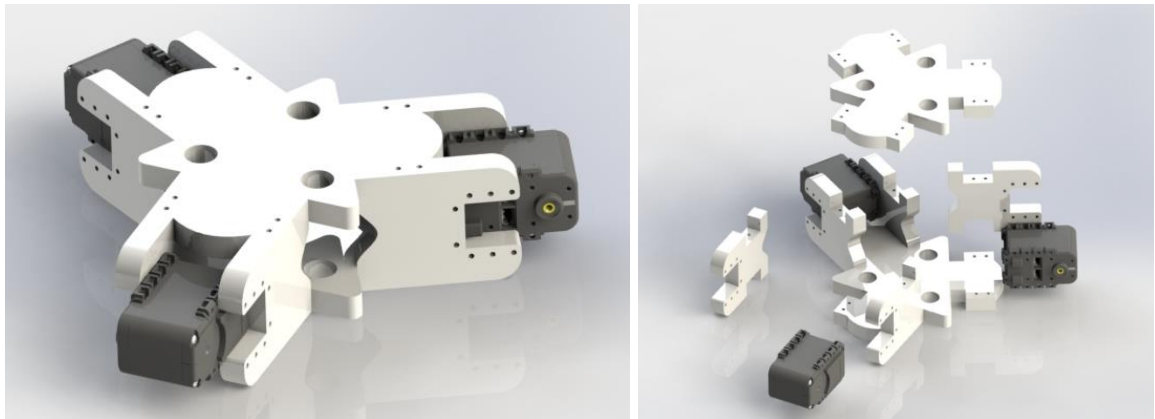


Fuente: Robotics, Motor paso a paso, 2016. [En línea]. ROBOTIS, [Consultado Mayo 2016] Disponible en internet: <http://robotics.shiftactive.com/lab/motores-electricos/>

El control de este motor se realiza mediante el mismo controlador de los motores de los brazos (OpenCM). Esto debido a que el motor sólo rota al llegar a la posición dada, por tanto, no interviene en el procesamiento de los demás motores.

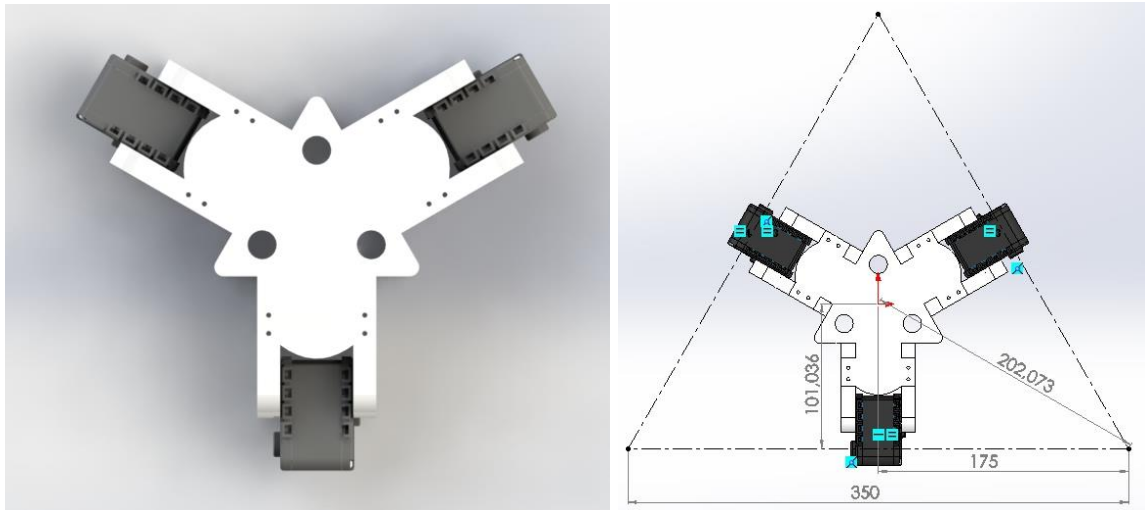
8.1.4. Diseño de base. Para empezar con el diseño estructural, se parte de la concepción de la base, elemento que será el apoyo fijo superior del robot, sosteniendo los motores, brazos superiores, brazos inferiores y efector final. La base se ha diseñado en función a los motores escogidos estudiados en el apartado anterior, por tanto, se ha buscado adaptar su forma con respecto al modo de sujeción con ellos. Sin embargo, se ha respetado lo que corresponde al dimensionamiento establecido en la tabla 3. A continuación un bosquejo de lo que se pretende.

Figura 49. Vista isométrica y en explisionado, base superior con motores.



Fuente: Elaboración propia.

Figura 50. Vista superior de la base y medidas



Fuente: Elaboración propia.

8.1.5. Diseño de brazos y articulaciones. Se han diseñado en primer lugar los brazos superiores, también en función de la forma de sujeción a los motores, quienes se conectan de manera directa, sin ningún acople mecánico especial. La longitud de dichos brazos respeta lo determinado en la tabla 3, como se ilustra en la Figura 51.

Figura 51. Modelo de brazos superiores.



Fuente: Elaboración propia.

Figura 52. Vista superior de brazos.



Fuente: Elaboración propia.

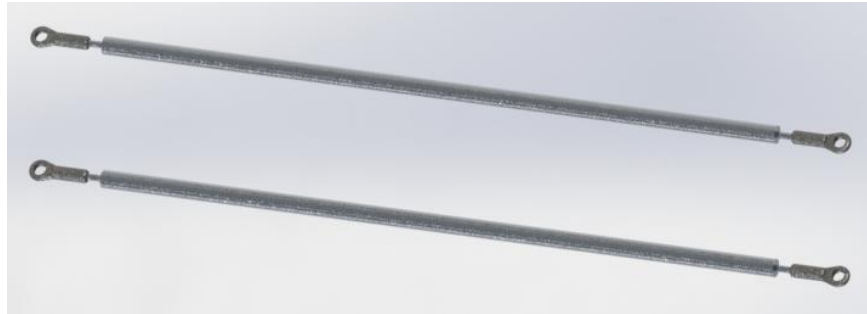
Figura 53. Vista explosionada.



Fuente: Elaboración propia.

Los brazos inferiores, correspondientes a un sistema de barras paralelas se han diseñado bajo la concepción de articular cada esquina mediante rotulas como lo ilustra la Figura 54. Se observa que son totalmente móviles, y su adaptación a los brazos superiores y el efector final es muy similar.

Figura 54. Modelo de brazos inferiores.



Fuente: Elaboración propia.

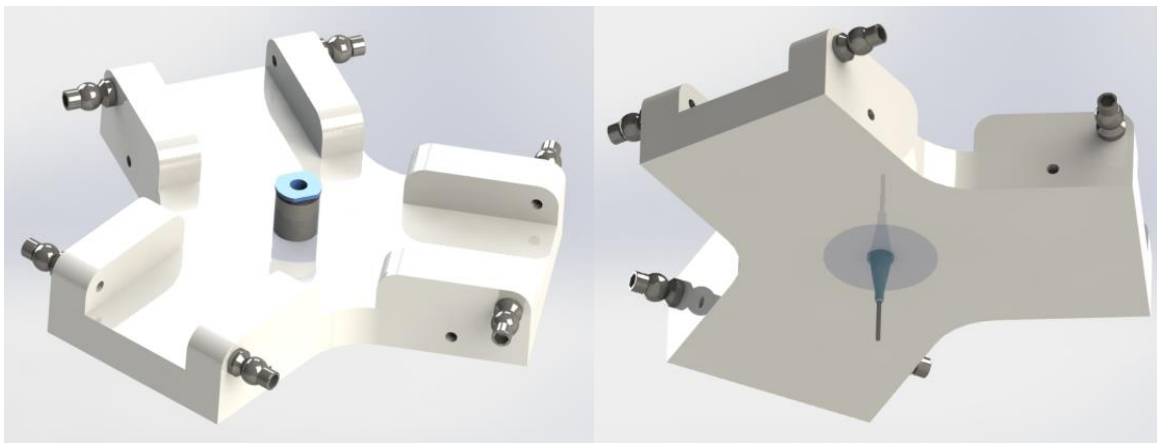
Figura 55. Modelo de la rótula.



Fuente: Elaboración propia.

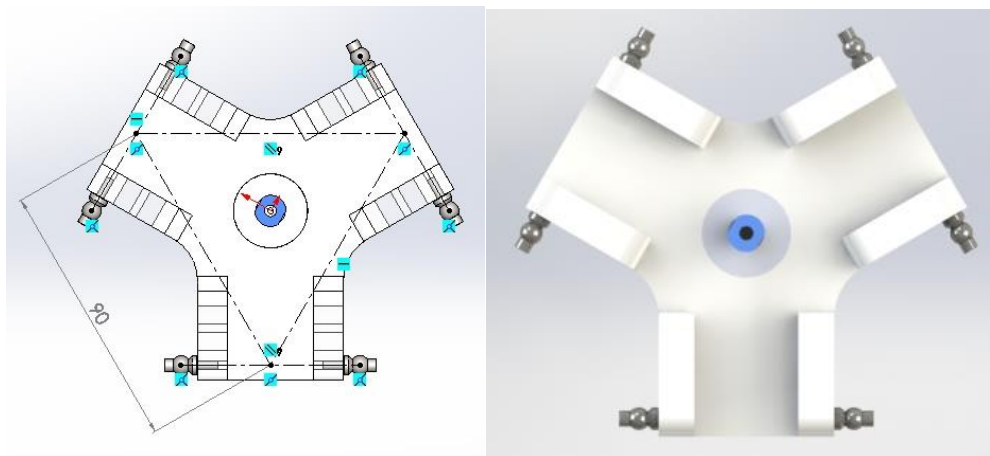
8.1.6. Diseño del TCP y Ventosa. En esta fase, se tiene en cuenta como aspecto importante, al motor que da rotación a la ventosa, pues este debe estar acoplado de manera fija, pero permitiendo la movilidad de la ventosa con su respectiva manguera neumática.

Figura 56. Modelo del TCP.



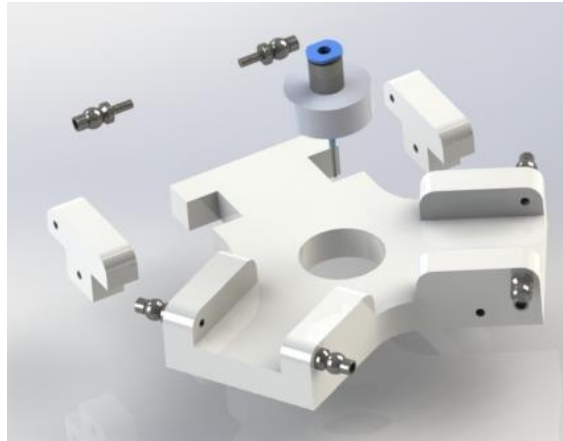
Fuente: Elaboración propia.

Figura 57. Vista superior del TCP y medidas.



Fuente: Elaboración propia.

Figura 58. Vista explosionada del TCP.

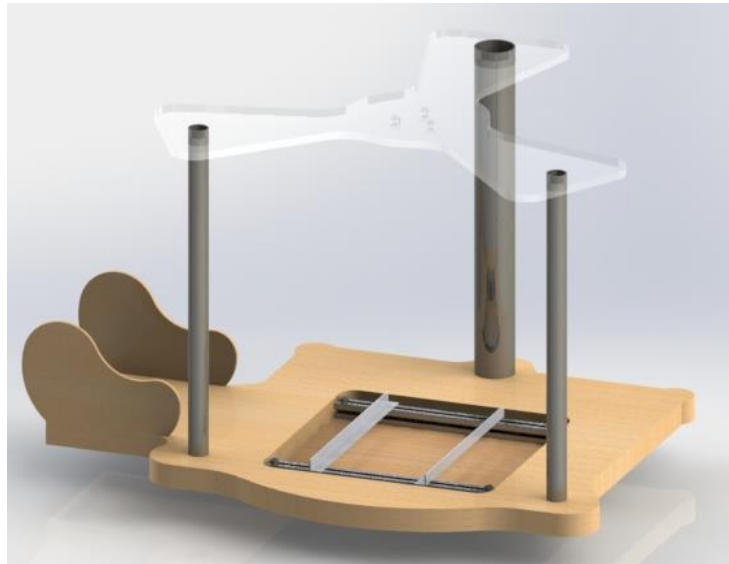


Fuente: Elaboración propia.

8.1.7. Soporte estructural. Finalmente, se diseña el soporte que sostiene todo el robot.

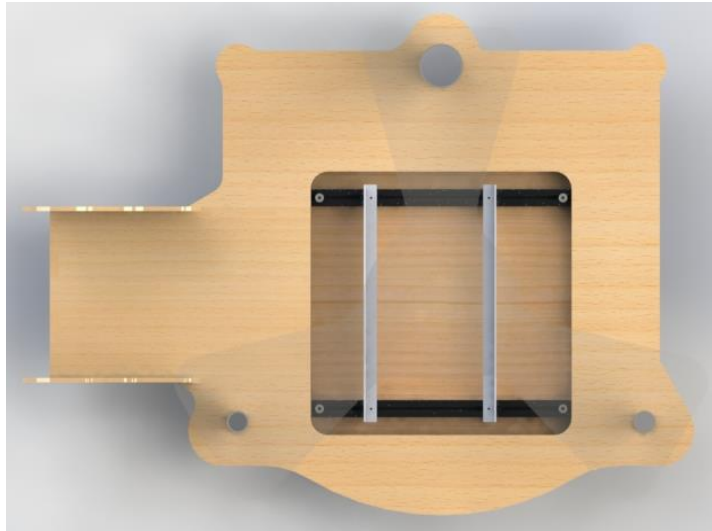
Este contempla además la zona donde la PCB es asegurada y la zona de las portacintas (izquierda).

Figura 59. Modelo del Soporte.



Fuente: Elaboración propia.

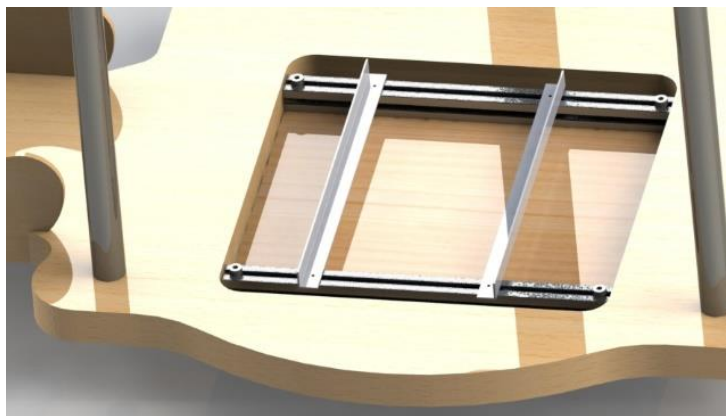
Figura 60. Vista superior del Soporte.



Fuente: Elaboración propia.

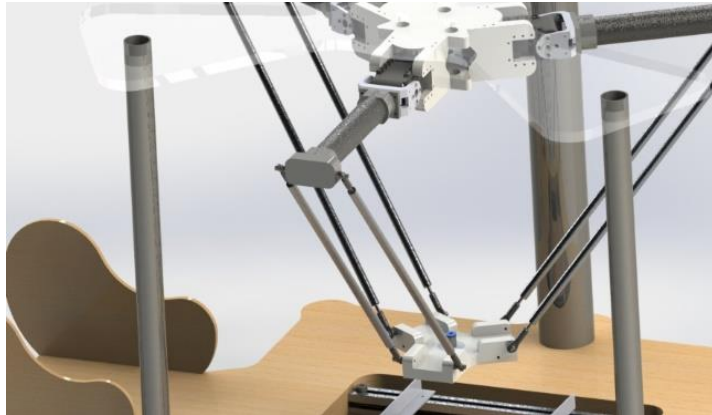
Se observa en la siguiente Figura, la zona donde se sujeta la PCB. El sistema se realiza mediante dos ángulos que se aseguran mediante dos guías perpendiculares, realizando el aseguramiento mediante tornillos.

Figura 61. Detalle de zona de sujeción de PCB.



Fuente: Elaboración propia.

Figura 62. Detalle de brazos articulados.



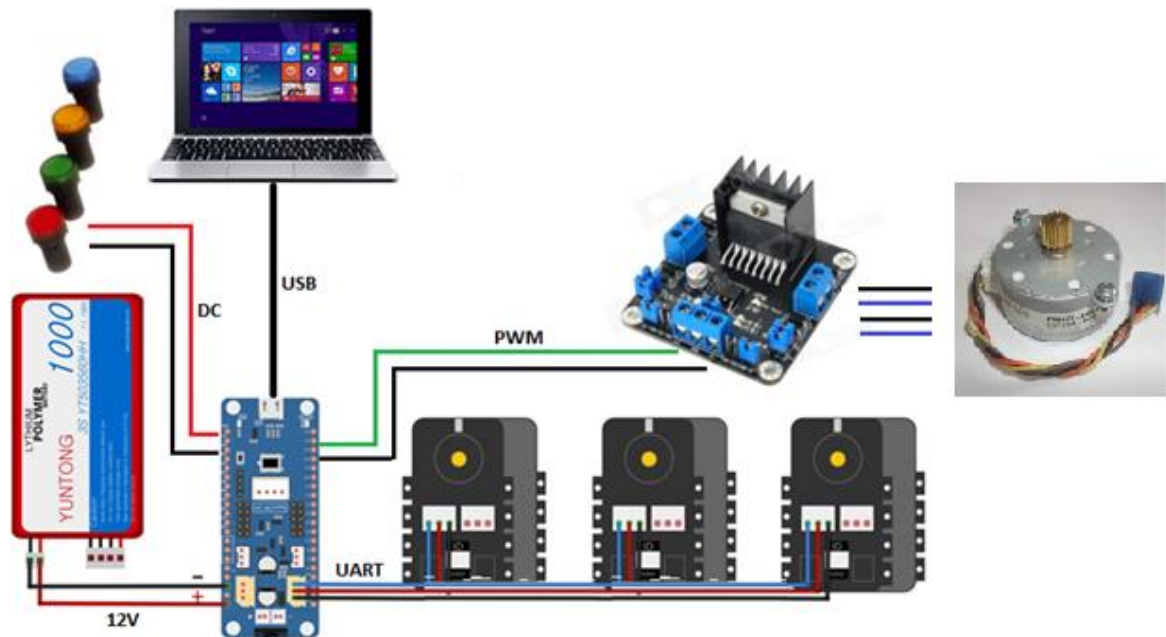
Fuente: Elaboración propia.

8.2. DISEÑO DE SISTEMA ELECTRÓNICO

El sistema electrónico debe ser capaz de procesar las señales que provienen del computador que ejecuta la cinemática, y enviar los datos correspondientes a la posición de los motores. Como se había mencionado, los motores poseen un sistema de comunicación serial que permite su manipulación y control mediante conexión Half Duplex. Un aspecto importante en este punto es la velocidad de procesamiento y velocidad de transmisión, aspectos que se deben considerar en el momento de seleccionar el controlado adecuado,

8.2.1. Planteamiento. La idea base radica en el hecho de manipular los tres motores de los brazos mediante el controlador OpenCM (concepto 3) a través del puerto USART de la misma. En la Figura 63 se observa la presencia del cuarto motor, correspondiente a la rotación de la ventosa. La conexión se hace a través de un drive genérico que recibe el dato mediante PWM, y éste lo traduce a pasos del motor.

Figura 63. Planteamiento del sistema electrónico.



Fuente: Elaboración propia.

8.2.2. Selección de sistema electrónico embebido. De acuerdo al concepto 3, el controlador del sistema es el sistema embebido OpenCM9.04. Este controlador de marca Robotis, está diseñado expresamente para el control de motores de dicha marca. Posee cuatro puertos de comunicación serial, en donde se realiza la conexión hacia los motores. Cabe destacar que el modo de conexión de los motores se realiza en modo bus. La Figura 65 ilustra el modo básico de conexión.

Figura 64. Controlador OpenCM9.04



Fuente: ROBOTIS, OpenCM9.04 Manual, 2010. [En línea]. support.robotis [Consultado marzo 2016] Disponible en internet: <http://support.robotis.com/en/product/auxdevice/controller/opencm9.04.htm>

Figura 65. Modo de conexión básico hacia motor.



Fuente: ROBOTIS, OpenCM9.04 Manual, 2010. [En línea]. support.robotis [Consultado Marzo 2016] Disponible en internet: <http://support.robotis.com/en/product/auxdevice/controller/opencm9.04.htm>

Como se observa en la Figura 66, de la placa se conecta un motor, este a su vez se conecta al siguiente y de la misma manera al último.

Figura 66. Conexión tipo bus mediante puerto serial.



Fuente: Elaboración propia.

8.2.3. Programación de sistema embebido. El controlador en mención se programa mediante lenguaje C, cuya plataforma es idéntica a la plataforma que implementa arduino.

La base de la programación se ha realizado en una interrupción en el puerto serial (USB) integrado. Cuando esto sucede, se reciben los datos correspondientes a velocidad y posición.

Pero la recepción se realiza de manera protocolizada, es decir, primero se recibe un dato correspondiente a la acción (ejemplo: establecer posición, o establecer velocidad), luego la información (ejemplo: posición o velocidad angular), y finalmente un dato adicional que se usa para la rotación del motor de la ventosa, cuando es necesario.

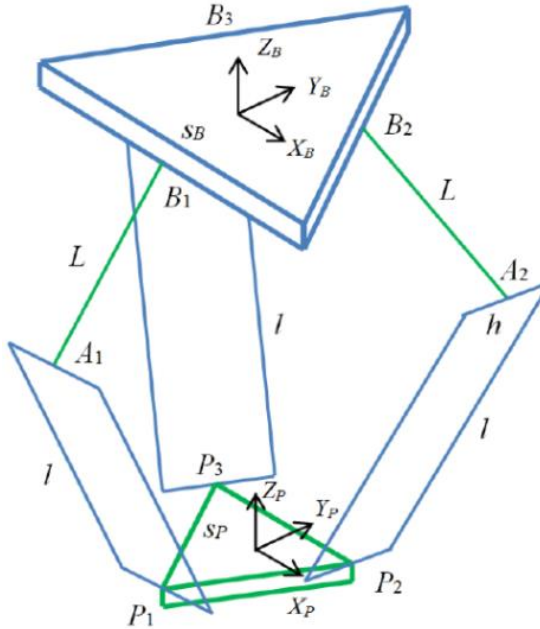
Para la lectura de las variables de los motores como la temperatura, la carga, la posición actual y velocidad actual, la trama es parecida, siendo el primer dato, la acción de leer, seguido de el parámetro que corresponde a lo que se va a leer de manera puntual (ejemplo: carga del motor1).

8.3. DISEÑO DE SISTEMA DE CONTROL

El sistema de control del robot se realiza con base en la cinemática inversa del robot y la generación de las trayectorias, que para el presente caso se ha elegido trabajar con interpolaciones cúbicas. Por otro lado, no ha sido necesario implementar una acción de control realimentada, como lo sería un PID por ejemplo, y esto debido a que los motores internamente poseen un sistema electrónico que ejecuta el control bien sea de posición, velocidad o fuerza. Por tanto, el sistema de control se restringirá a la generación adecuada de las trayectorias, teniendo en cuenta la cinemática inversa del robot.

8.3.1. Geometría del robot. Para empezar, es importante rescatar los datos geométricos del robot. La Figura 67 ilustra la estructura con las variables a relacionar mediante matemática vectorial.

Figura 67. Variables geométricas del robot.



Fuente: R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, 2016. [En línea]. ohio. [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

La ecuación que modela los vectores de los brazos viene dada de la siguiente manera:

$$\{^B_i B\} + \{^B_i L\} + \{^B_i l\} = \{^B_P P\} + [^B_P R] \{^P_i P\} = \{^B_P P\} + \{^P_i P\} \quad i = 1, 2, 3$$

En la ecuación se observa la igualdad, donde se pretende mostrar de manera vectorial, la representación del TCP.

8.3.2. Obtención de la cinemática inversa. El estudio de la cinemática inversa tiene como objetivo encontrar una ecuación que sea función de la posición deseada dentro de un volumen de trabajo, y como resultado arroje las posiciones angulares de los motores. Para empezar, es necesario definir las ecuaciones modelo. Por otro lado, y teniendo en cuenta que el robot Delta es de tipo Paralelo, el método para resolver la cinemática es más complejo que un robot en Serie, donde se puede hallar mediante métodos geométricos. El método de obtención a aplicar será mediante ecuaciones vectoriales que definan la posición de la Base (origen de coordenadas) y el TCP (coordenadas deseadas).

Como se observa en la Figura 67, los brazos inferiores, denotados con l , están conectados al TCP, por tanto, obtener la posición final de l en las tres cadenas cinemáticas será necesario para hallar la cinemática.

Para que se entienda el proceso de manera práctica, lo que se hace en teoría es ubicar tres esferas en el punto entre L y l (entre brazos superiores e inferiores) y la intersección dará como resultado la posición deseada. Esto es posible debido a que L es conocido, el cual viene definido espacialmente por el ángulo del motor, sin embargo, l no es definido dado que corresponde a un paralelogramo.

A continuación, se define vectorialmente l (brazos inferiores):

$$l_i = \|\{^B l\}\| = \|\{^B P\} + \{^P l\} - \{^B B\} + \{^B L\}\| \quad i = 1, 2, 3$$

Ahora, como el cuadrado del valor absoluto del vector corresponde a sus componentes x , y , z , otra ecuación resulta:

$$l_i^2 = \|\{^B l\}\|^2 = l_{ix}^2 + l_{iy}^2 + l_{iz}^2$$

Como el vector $\{^B P\} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$, define las variables cartesianas, se procede a definir B_i y P_i :

$${}^B B_1 = \begin{Bmatrix} 0 \\ -wB \\ 0 \end{Bmatrix} \quad {}^B B_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}wB \\ \frac{1}{2}wB \\ 0 \end{Bmatrix} \quad {}^B B_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}wB \\ \frac{1}{2}wB \\ 0 \end{Bmatrix}$$

$${}^B L_1 = \begin{Bmatrix} 0 \\ -L\cos\theta_1 \\ -L\sin\theta_1 \end{Bmatrix} \quad {}^B L_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}L\cos\theta_2 \\ \frac{1}{2}L\cos\theta_2 \\ -L\sin\theta_2 \end{Bmatrix} \quad {}^B L_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}L\cos\theta_3 \\ \frac{1}{2}L\cos\theta_3 \\ -L\sin\theta_3 \end{Bmatrix}$$

Se sustituye la ecuación de las componentes I en la ecuación de L y se obtiene:

$${}^B I_1 = \begin{Bmatrix} x \\ y + L\cos\theta_1 + a \\ z + L\sin\theta_1 \end{Bmatrix} \quad {}^B I_2 = \begin{Bmatrix} x - \frac{\sqrt{3}}{2}L\cos\theta_2 + b \\ y - \frac{1}{2}L\cos\theta_2 + c \\ z + L\sin\theta_2 \end{Bmatrix} \quad {}^B I_3 = \begin{Bmatrix} x + \frac{\sqrt{3}}{2}L\cos\theta_3 - b \\ y - \frac{1}{2}L\cos\theta_3 + c \\ z + L\sin\theta_3 \end{Bmatrix}$$

Siendo:

$$a = wB - uP$$

$$b = \frac{sP}{2} - \frac{\sqrt{3}}{2}wB$$

$$c = wP - \frac{wB}{2}$$

Y se obtienen finalmente las tres ecuaciones del lazo cinemático:

$$\begin{aligned} 2L(y+a)\cos\theta_1 + 2zL\sin\theta_1 + x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 &= 0 \\ -L(\sqrt{3}(x+b) + y+c)\cos\theta_2 + 2zL\sin\theta_2 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 &= 0 \\ L(\sqrt{3}(x-b) - y-c)\cos\theta_3 + 2zL\sin\theta_3 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 &= 0 \end{aligned}$$

Se observa que son tres ecuaciones con tres incógnitas, por tanto, tiene solución.

Antes de proceder, es importante definir las ecuaciones de los puntos A y B

(correspondientes a los brazos superiores) ${}^B \mathbf{A}_i = {}^B \mathbf{B}_i + {}^B \mathbf{L}_i, i=1,2,3$.

$${}^B \mathbf{A}_1 = \begin{Bmatrix} 0 \\ -w_B - L \cos \theta_1 \\ -L \sin \theta_1 \end{Bmatrix} \quad {}^B \mathbf{A}_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}(w_B + L \cos \theta_2) \\ \frac{1}{2}(w_B + L \cos \theta_2) \\ -L \sin \theta_2 \end{Bmatrix} \quad {}^B \mathbf{A}_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}(w_B + L \cos \theta_3) \\ \frac{1}{2}(w_B + L \cos \theta_3) \\ -L \sin \theta_3 \end{Bmatrix}$$

Con estos datos, se define una ecuación de la forma:

$$E_i \cos \theta_i + F_i \sin \theta_i + G_i = 0 \quad i=1,2,3$$

Donde al despejar se obtiene:

$$E_1 = 2L(y+a)$$

$$F_1 = 2zL$$

$$G_1 = x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2$$

$$E_2 = -L(\sqrt{3}(x+b) + y + c)$$

$$F_2 = 2zL$$

$$G_2 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(xb + yc) - l^2$$

$$E_3 = L(\sqrt{3}(x-b) - y - c)$$

$$F_3 = 2zL$$

$$G_3 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(-xb + yc) - l^2$$

Resolviendo el conjunto de ecuaciones obtenemos la posición, por tanto, optamos por solucionar mediante la tangente del ángulo medio:

$$t_i = \tan \frac{\theta_i}{2}$$

Se definen las siguientes relaciones trigonométricas

$$\cos \theta_i = \frac{1-t_i^2}{1+t_i^2} \quad \sin \theta_i = \frac{2t_i}{1+t_i^2}$$

Sustituyendo en la ecuación del ángulo medio se obtiene:

$$E_i \left(\frac{1-t_i^2}{1+t_i^2} \right) + F_i \left(\frac{2t_i}{1+t_i^2} \right) + G_i = 0 \quad E_i(1-t_i^2) + F_i(2t_i) + G_i(1+t_i^2) = 0$$

Que se puede resolver mediante una ecuación cuadrática:

$$(G_i - E_i)t_i^2 + (2F_i)t_i + (G_i + E_i) = 0 \quad t_{i,2} = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i}$$

Finalmente obtenemos la ecuación de la cinemática inversa:

$$\theta_i = 2 \tan^{-1}(t_i)$$

Siendo ti el resultado de la ecuación cuadrática.

Implementación en MatLab:

```
%% FUNCION QUE DESARROLLA LA CINEMÁTICA INVERSA
function sp=ipkDelta(x,y,z)
    a=wB-uP;
    b=(sP/2)-(sqrt(3)/2)*wB;
    c=wP-0.5*wB;

    E1=2*L*(y+a);
    F1=2*z*L;
    G1=x^2+y^2+z^2+a^2+L^2+2*y*a-1^2;

    E2=-L*(sqrt(3)*(x+b)+y+c);
    F2=2*z*L;
    G2=x^2+y^2+z^2+b^2+c^2+L^2+2*(x*b+y*c)-1^2;

    E3=L*(sqrt(3)*(x-b)-y-c);
    F3=2*z*L;
    G3=x^2+y^2+z^2+b^2+c^2+L^2+2*(-x*b+y*c)-1^2;

    function qf=ti(E,F,G)
        ta=-(F+(sqrt(E^2+F^2-G^2)))/(G-E);
        tb=-(F-(sqrt(E^2+F^2-G^2)))/(G-E);
        qf=ta;
    end
    theta1=2*atan(ti(E1,F1,G1));
    theta2=2*atan(ti(E2,F2,G2));
    theta3=2*atan(ti(E3,F3,G3));
    sp=[theta1*180/pi theta2*180/pi theta3*180/pi]';
end
```

8.3.3. Generación de trayectorias. Uno de los aspectos que afectan directamente la estructura en relación al movimiento, es la forma en como las articulaciones se mueven dentro del espacio de trabajo. Trayectorias bruscas pueden desestabilizar la máquina, provocando errores de calibración hasta llegar al deterioro y desajuste de los componentes mecánicos. Es por esto que las trayectorias que realiza el efector final deben ser supervisadas.

La función de un generador de trayectorias es generar las entradas de referencia para el sistema de control de movimiento, que se encarga de garantizar que el robot ejecute el recorrido planificado en el tiempo planificado.

Las trayectorias pueden especificarse:

En espacio cartesiano: pasar de la localización $L1 = (x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1)$ a la localización $L2 = (x_2, y_2, z_2, \alpha_2, \beta_2, \gamma_2)$.

En espacio de articulaciones: pasar de la configuración de variables de articulación $Q1 = (q_{11}, q_{21}, q_{31}, q_{41}, q_{51})$ a otra $Q2 = (q_{12}, q_{22}, q_{32}, q_{42}, q_{52})$.

La generación de trayectorias en espacio de articulaciones es más sencilla, y permite generar funciones lineales para cada uno de los ejes (por ejemplo, trapezoides). Sin embargo, genera un movimiento no lineal del efector final.

Si la trayectoria se genera en espacio cartesiano, el movimiento del efector final será lineal, lo que provoca que cada articulación realice un movimiento no-lineal. La generación de trayectorias en espacio cartesiano es más cómoda para el usuario, ya que puede ver el recorrido que realizará el efector final. El inconveniente principal es que hay que resolver la cinemática inversa para cada punto de la trayectoria, con el consiguiente coste computacional. Este coste es una de las causas de que la mayoría de los robots industriales funcionen de 3 a 4 veces más despacio cuando generan trayectorias cartesianas a cuando lo hacen en espacio de articulaciones.

Uno de los métodos comunes de generar trayectorias en espacio de articulaciones es el movimiento punto a punto. En este tipo de movimiento se utilizan polinomios cúbicos para definir la trayectoria. La forma mínima de hacerlo es mediante

polinomios de tercer grado. La posición de la articulación se determina mediante el siguiente polinomio:

$$q(t) = at^3 + bt^2 + ct + d$$

El perfil de la velocidad y la aceleración se obtiene derivando:

$$\dot{q}(t) = 3at^2 + 2bt + c \quad \text{Primera derivada}$$

$$\ddot{q}(t) = 6at + 2b \quad \text{Segunda derivada}$$

Para calcular los coeficientes del polinomio partimos de la posición inicial q_i y la posición final q_f . Sabemos que el polinomio en $t=0$ (principio de la trayectoria) deberá darnos la posición inicial y que el polinomio en $t=1$ (final de la trayectoria) deberá dar la posición final, con lo que definimos 2 ecuaciones:

$$\text{I.} \quad q(0) = d = q_i$$

$$\text{II.} \quad q(1) = a + b + c + d = q_f$$

También sabemos que la velocidad al principio y al final de la trayectoria deberá ser cero, con lo que podemos definir 2 ecuaciones más:

$$\text{III.} \quad \dot{q}(0) = c = 0$$

$$\text{IV.} \quad \dot{q}(1) = 3a + 2b + c = 0$$

De estas cuatro ecuaciones podemos obtener los cuatro coeficientes del polinomio para cada articulación.

Implementación en MatLab:

```

%%  GENERADOR DE TRAYECTORIAS (ESPACIO DE LAS ARTICULACIONES)
function Tray=genTray(qi,qf,tf)
    N=length(qi);
    Tray=[];
    for i=1:N
        a(i)=2*(qi(i)-qf(i))/tf^3;
        b(i)=(-3*a(i)*tf)/2;
        c(i)=0;
        d(i)=qi(i);
    end
    for t=0:0.1:tf
        for i=1:N
            q(i)=a(i)*t^3 + b(i)*t^2 + c(i)*t + d(i);
        end
        Tray=[Tray;q];
    end
end

```

El Anexo H muestra el código MatLab completo, con un ejemplo en donde se puede observar el movimiento del robot en 7 posiciones definidas.

```

%%  GENERADOR DE TRAYECTORIAS (ESPACIO CARTESIANO)
function Tray=genTray2(pi,pf,tf)
    N=length(pi);
    Tray=[];
    for i=1:N
        a(i)=2*(pi(i)-pf(i))/tf^3;
        b(i)=(-3*a(i)*tf)/2;
        c(i)=0;
        d(i)=pi(i);
    end
    for t=0:0.1:tf
        for i=1:N
            p(i)=a(i)*t^3 + b(i)*t^2 + c(i)*t + d(i);
        end
        Tray=[Tray;p];
    end
end

```


8.4. DISEÑO DE SISTEMA INFORMÁTICO

Esta última fase del proyecto contempla el desarrollo de una aplicación informática que permita:

- Leer los archivos de coordenadas de la PCB a ensamblar.
- Ejecutar las trayectorias mediante las ecuaciones cinemáticas.
- Enviar por puerto serial la información de posición de los motores.
- Leer el estado de los motores.

8.4.1. Requisitos y requerimientos. En primera instancia, la aplicación debe ser Open Software, escrito en un lenguaje de programación orientado a objetos y compatible con el sistema operativo Windows 7, 8, 8.1 y 10.

Se debe contemplar además el uso de librerías dinámicas que permitan usar los puertos COM del computador, de tal manera que sea fácil la comunicación al sistema electrónico.

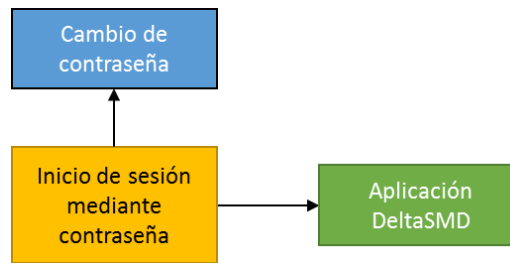
Finalmente, el programa debe trabajar de manera sincronizada al sistema electrónico y la ejecución de las trayectorias mediante el modelamiento cinemático debe poderse realizar en tiempo real.

8.4.2. Selección de plataforma de programación. De acuerdo a lo anterior, se escoge la el lenguaje de programación JAVA, por su gran parecido al lenguaje C++, su practicidad en el manejo de clases, métodos y objetos, y porque no requiere licencia para uso ni distribución.

Por otro lado, el entorno de programación escogido es NetBeans por su enorme practicidad en el manejo de proyectos de múltiples archivos.

8.4.3. Interfaz gráfica de usuario (GUI). Se desarrollaron tres interfaces gráficas como lo ilustra la Figura 68. Considerando la seguridad de la operación de la máquina, al iniciar la aplicación, se debe ingresar un nombre de usuario y contraseña, los cuales pueden ser modificados mediante una contraseña de administrador.

Figura 68. Interfaces de usuario.



Fuente: Elaboración propia.

Cuando se ejecuta la aplicación, se despliega una interfaz gráfica como lo ilustra la Figura 69.

Figura 69. Interfaz gráfica de usuario: Inicio de sesión.



Fuente: Elaboración propia.

Un administrador tiene permitido realizar el cambio de contraseñas mediante una clave principal, desplegándose una interfaz como lo ilustra la Figura 70.

Figura 70. Interfaz para cambio de contraseñas y administración de usuarios.

The screenshot shows a window titled 'Comunicacion RS-232' with a green title bar. Inside, there's a section titled 'Administrador de Usuarios'. Under 'Usuarios:', a list box contains '1. "armando.uribe"'. Below this is a 'Contraseña:' field with the value '2116676'. Three buttons are present: 'Eliminar' (highlighted with a dashed border), 'Cambiar Nombre', and 'Cambiar Contraseña'. Below these is a 'Crear Usuario' section with 'Nombre de Usuario:' and 'Contraseña' input fields, and 'Crear' and 'Limpiar Campos' buttons. At the bottom are 'Volver', 'Cambiar Contraseña Administrador', and 'Salir' buttons.

Fuente: Elaboración propia.

Una vez se han digitado las credenciales de ingreso, se despliega la aplicación principal (ver Figura 71). Esta tiene cinco paneles correspondientes a:

- Conexión
- Comander
- Programa
- Control
- Logger

Figura 71. Interfaz de aplicación principal.

The interface is titled "DeltaSMD | IPK Control". It is divided into several functional areas:

- Conexión (Connection):** Includes a "Puerto:" dropdown menu, an "Abrir Conexión" (Open Connection) button, and a "Cerrar Conexión" (Close Connection) button.
- Comander (Command):** Features a "Comando a enviar:" (Command to send) text input field with an "Enviar Dato" (Send Data) button, and a "Comando recibido:" (Command received) text input field with a "Recibir Dato" (Receive Data) button.
- Programa (Program):** Contains fields for "PCB:", "Nombre:" (Name), "Tamaño:" (Size), and "Lineas:" (Lines). Below these is a "Cargar PCB" (Load PCB) button. At the bottom of this section are four buttons: "Ejecutar Todo" (Execute All), "Ejecutar Linea" (Execute Line), "Ejecutar Selección" (Execute Selection), and "Eliminar Linea" (Delete Line).
- Control:** This section contains various control parameters:
 - FPK:** J1, J2, J3, each with "+" and "-" buttons.
 - Ángulo:** A field with a degree symbol and a "Set" button.
 - Velocidad:** A field with "°/s" and a "Set" button.
 - Avance Motor:** A field with a degree symbol and a "Set" button.
 - Avance Lineal:** A field with "mm" and a "Set" button.
 - Home:** A field with a "Set" button.
 - TCP:** "ON" and "OFF" buttons.
 - PCB R 1:** A field with a "Set" button.
 - PCB R 2:** A field with a "Set" button.
 - IPK:** X, Y, Z, each with "+" and "-" buttons.
 - Posición TCP:** Fields for X, Y, and Z in "mm", each with a "Set" button.
 - Ref1, Ref2, Ref3:** Fields with "Set" buttons.
- Logger:** A large text area for logging, with "Limpiar Log" (Clear Log) and "Salir" (Exit) buttons at the bottom.

Fuente: Elaboración propia.

➤ **Panel Conexión:**

Dentro de este panel se tienen las funciones necesarias para establecer la comunicación entre la aplicación y el sistema electrónico mediante puerto USB (emulado como serial).

➤ **Panel Comander:**

Este panel contiene dos opciones, enviar o recibir un dato. Se implementó con el objetivo de realizar envíos manuales al sistema electrónico, realizar pruebas y ejecutar acciones en modo administrador.

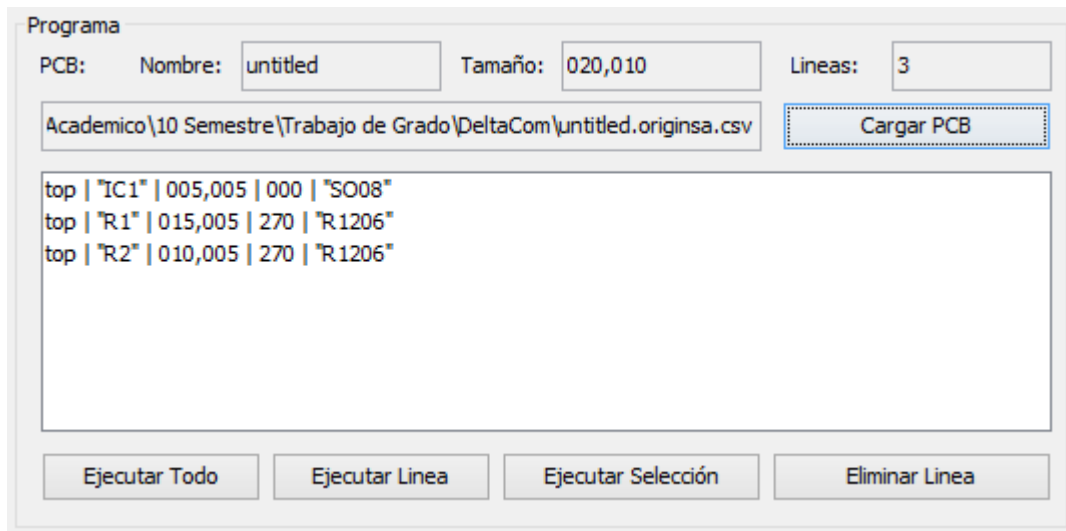
➤ **Panel Logger:**

En este panel se visualiza el estado de la aplicación, los estados y alarmas.

➤ **Panel Programa:**

El panel programa permite abrir un archivo de coordenadas (generado por Eagle o Altium) y separarlo de tal manera que se puedan visualizar todos los componentes, su ubicación, el tipo de dispositivo, orientación y etiqueta. La Figura 72 ilustra la posibilidad de ejecutar el posicionamiento línea a línea, por selecciones, o todo completo. También se permite eliminar líneas con el objetivo de omitir componentes cuando sea necesario.

Figura 72. Panel de programa.



Fuente: Elaboración propia.

➤ **Panel Control:**

Este último panel permite realizar acciones directas sobre el robot, tales como movimientos en el espacio cartesiano, en el espacio articulado, activar el TCP (ventosa), establecer la velocidad de operación, o establecer referencias como el punto Home o las esquinas de la PCB.

8.4.4. Definición de clases, objetos y métodos. Se han definido además de las clases fundamentales; clases de interfaces y clases de apoyo, las siguientes correspondientes a la operación del robot:

- Delta
- genTray
- readPCB

La primera (Delta), es una clase que permite desarrollar la cinemática del robot:

```
public class Delta
{
    private double res=1.024/0.3;
    public double[] Result;
    public double[] Goal;
    private IPK IPK;
    public Delta()
    {
        Result = new double[3];
        Goal = new double[3];
        IPK = new IPK();
    }
    public void solveIp(double x, double y, double z)
    {
        Result=IPK.solveIPK(x, y, z);
        Goal[0]=Math.round(Result[0]*res+512);
        Goal[1]=Math.round(Result[1]*res+512);
        Goal[2]=Math.round(Result[2]*res+512);
    }
}
```

La clase genTray permite desarrollar las trayectorias de acuerdo a la posición deseada:

```
public class genTray
{
    public genTray()
    {
    }
    public double[][] Tray(double[] qi, double[] qf, double tf)
    {
        int N=3;
        tf*=10;
        double[] a = new double[N];
        double[] b = new double[N];
        double[] c = new double[N];
        double[] d = new double[N];
        double[] q = new double[N];
        double[][] Tray = new double[(int)tf+1][N];
        for (int i=0;i<N;i++)
        {
            a[i]=2*(qi[i]-qf[i])/Math.pow(tf,3);
            b[i]=(-3*a[i]*tf)/2;
            c[i]=0;
            d[i]=qi[i];
        }
        for (int t=0;t<=tf;t++)
        {
            for (int i=0;i<N;i++)
            {
                q[i]=a[i]*Math.pow(t,3) + b[i]*Math.pow(t,2) + c[i]*t + d[i];
                Tray[t][i]=q[i];
            }
        }
        return Tray;
    }
}
```

Finalmente, la clase readPCB tiene la programación adecuada para leer un archivo de coordenadas y separarlo en diferentes partes:

```

public class readPCB
{
    String coordenadasLeidas;
    String[] coordenadasLeidasArr;
    String[] capa;
    String[] tipo;
    double[] posX;
    double[] posY;
    double[] rota;
    String[] tama;
    String contraseñaAdmin;
    String titulo;
    String area;
    double areaX;
    double areaY;
    String mensaje="";
    int tamaño;
    public readPCB(File archivo)
    {
        capa = new String[100];
        tipo = new String[100];
        posX = new double[100];
        posY = new double[100];
        rota = new double[100];
        tama = new String[100];
        try
        {
            FileReader fr = new FileReader(archivo);
            BufferedReader br = new BufferedReader(fr);
            titulo = br.readLine();
            area = br.readLine();
            area=area.split("Extents,",2)[1];
            areaX=Double.parseDouble(area.split(",")[0])*0.0254;
            areaY=Double.parseDouble(area.split(",")[1])*0.0254;
            while((coordenadasLeidas=br.readLine())!=null)
            {
                coordenadasLeidasArr = coordenadasLeidas.split(", ", 6);
                capa[tamaño] = coordenadasLeidasArr[0];
                tipo[tamaño] = coordenadasLeidasArr[1];
                posX[tamaño] = Double.parseDouble(coordenadasLeidasArr[2])*0.0254;
                posY[tamaño] = Double.parseDouble(coordenadasLeidasArr[3])*0.0254;
                rota[tamaño] = Double.parseDouble(coordenadasLeidasArr[4]);
                tama[tamaño] = coordenadasLeidasArr[5];
                tamaño++;
            }
            br.close();
            fr.close();
            mensaje="ok";
        } catch (FileNotFoundException ex)
        {
            mensaje="error";
        } catch (IOException ex)
        {
        }
    }
}

```


8.4.5. Comunicación serial. La comunicación serial al sistema electrónico se realiza mediante la siguiente clase:

```
public class Comunicacion
{
    private SerialIO puerto;

    public boolean establecerComunicacion(String port)
    {
        boolean e;
        puerto = new SerialIO();
        puerto.setPortName(port);
        puerto.setBaudRate(9600);
        e = puerto.open();
        puerto.write(48);
        return e;
    }

    public int getNumPuestos()
    {
        int i=0;
        try
        {
            for(String s : SerialIO.getPorts())
            {
                i++;
            }
        }
        catch(UnsatisfiedLinkError ex1)
        {
            JOptionPane.showMessageDialog(null, "No se encontro el archivo \"rxtxSerial.dll\"");
            System.exit(0);
        }
        return i;
    }

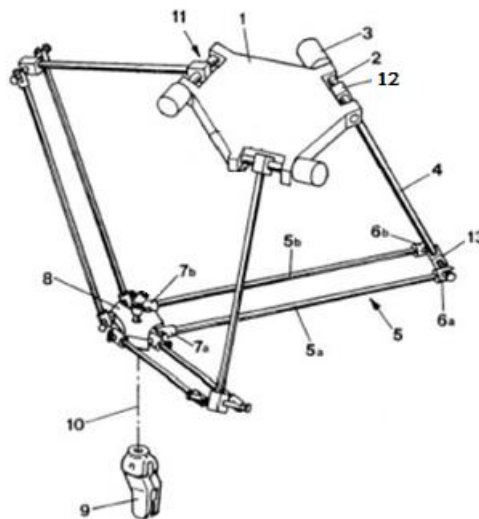
    public String[] getPuestos()
    {
        return SerialIO.getPorts();
    }
}
```

9. PROTOTIPADO

Con el objetivo de verificar la funcionalidad del robot bajo la integración de todos los subsistemas mecatrónicos desarrollados en el proceso de diseño, se elabora un prototipo de la máquina, en donde se ha podido emular su funcionalidad y la de todos los componentes principales y secundarios, analizando el flujo de los datos y la información digital que fluye desde el sistema informático, hasta los motores.

A continuación, se ilustran las piezas principales que conforman el robot. Partiendo del análisis de piezas, su función, características dinámicas y forma de conexión a la armadura del robot, se establece un patrón de ensamblaje, con el cual se garantiza una mayor eficiencia en el aprovechamiento del tiempo de ensamblaje, de acuerdo a los datos del análisis de tiempos.

Figura 73. Desglose de piezas del robot.



Fuente: R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, 2016. [En línea]. ohio [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

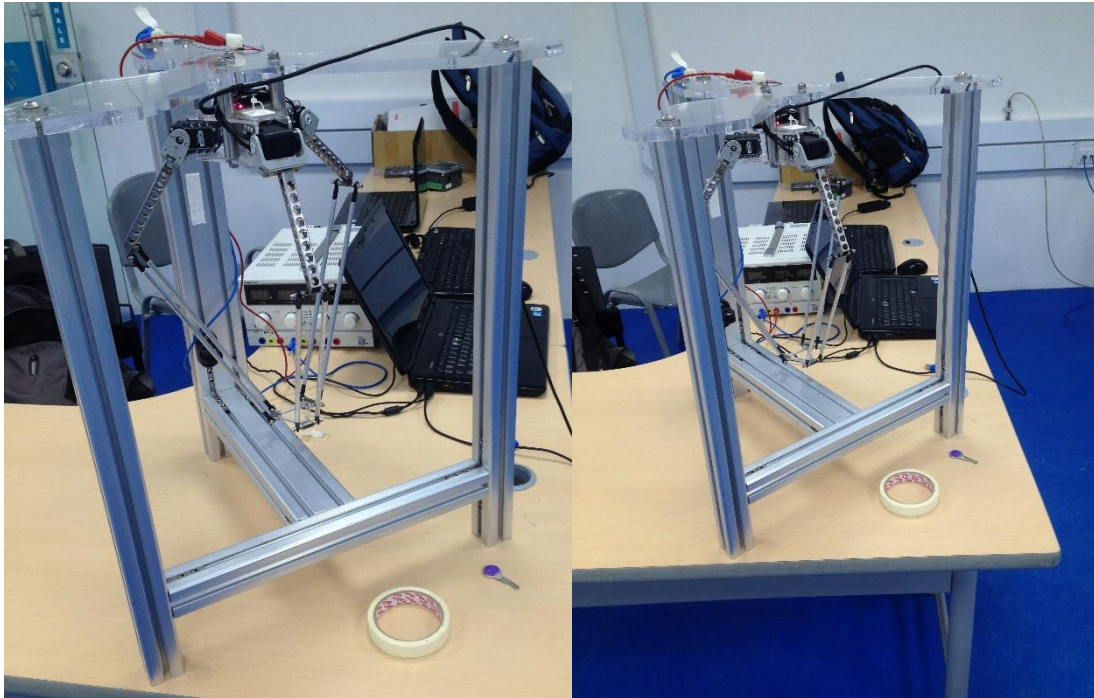
Tabla 19. Listado de partes del robot.

#	Componente
1	Robot Base
2	Acople Motor – Eje
3	Motor
4	Eje
5	Soporte Eje
6	Acople Eje – Soporte
7	Acople Soporte (Ventosa) - TCP
8	TCP
9	Ventosa
10	Eje Alineamiento
11	Acople Motor – Eje
12	Acople Motor – Eje
13	Acople Eje – Soporte

Fuente: Elaboración propia.

Cabe destacar que existen piezas cuya función es trivial como tornillos, tuercas, acoples, guías arandelas, etc. por lo cual sólo se mencionan los elementos cuya función es esencial en el ensamblaje del robot.

Figura 74. Prototipo del robot desarrollado.



Fuente: Elaboración propia.

A continuación, se ilustra en la tabla 14, la matriz de ensamblaje del robot Delta. En ella se pueden observar el listado de partes y el tiempo que tarda en realizar el ensamblaje de cada pieza

Tabla 20. Matriz de ensamblaje del robot Delta.

[illegible]

Fuente: Elaboración propia.

A continuación, el diagrama de ensamblaje del robot.

Figura 75. Diagrama de ensamble del robot.

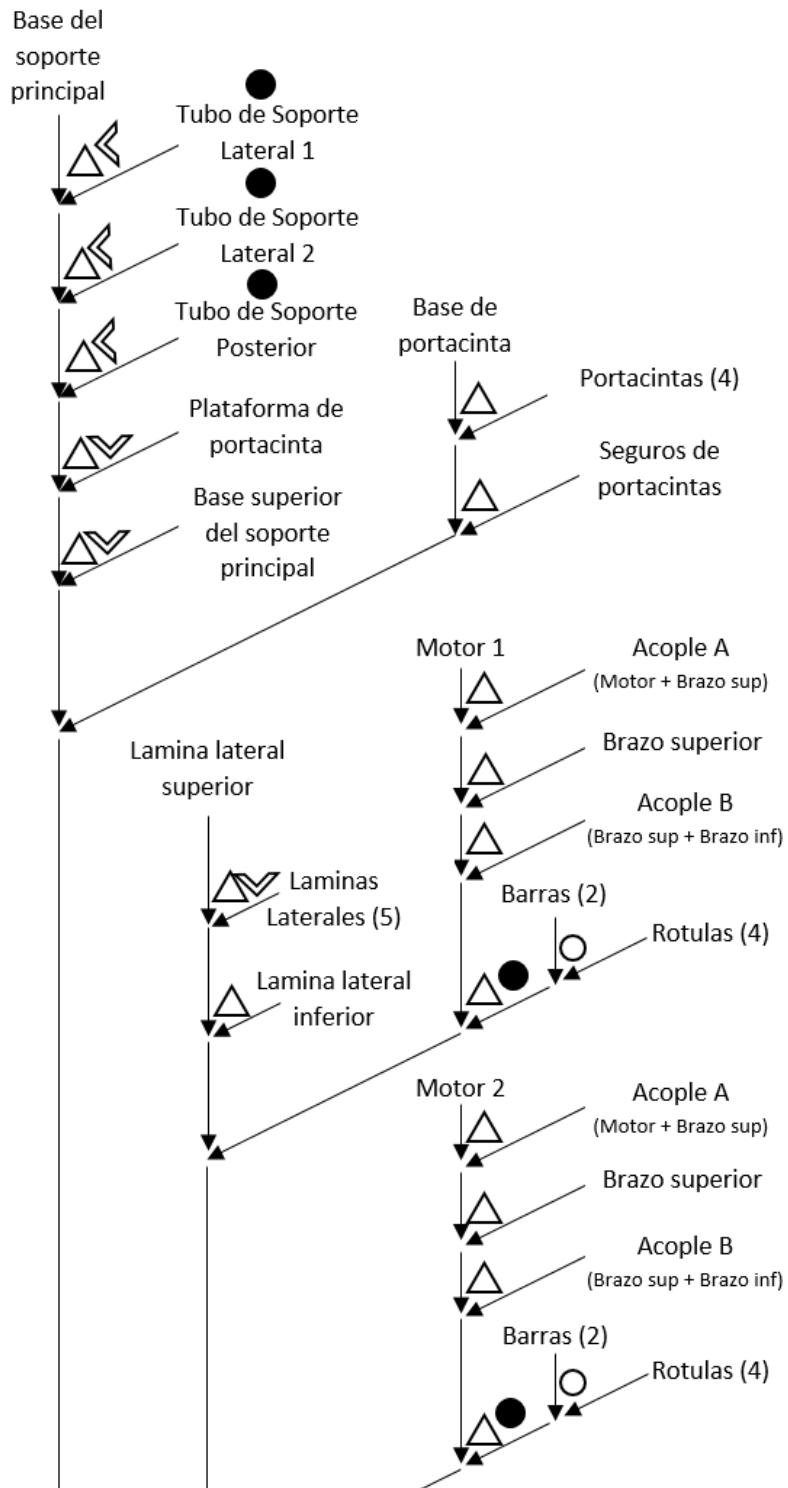
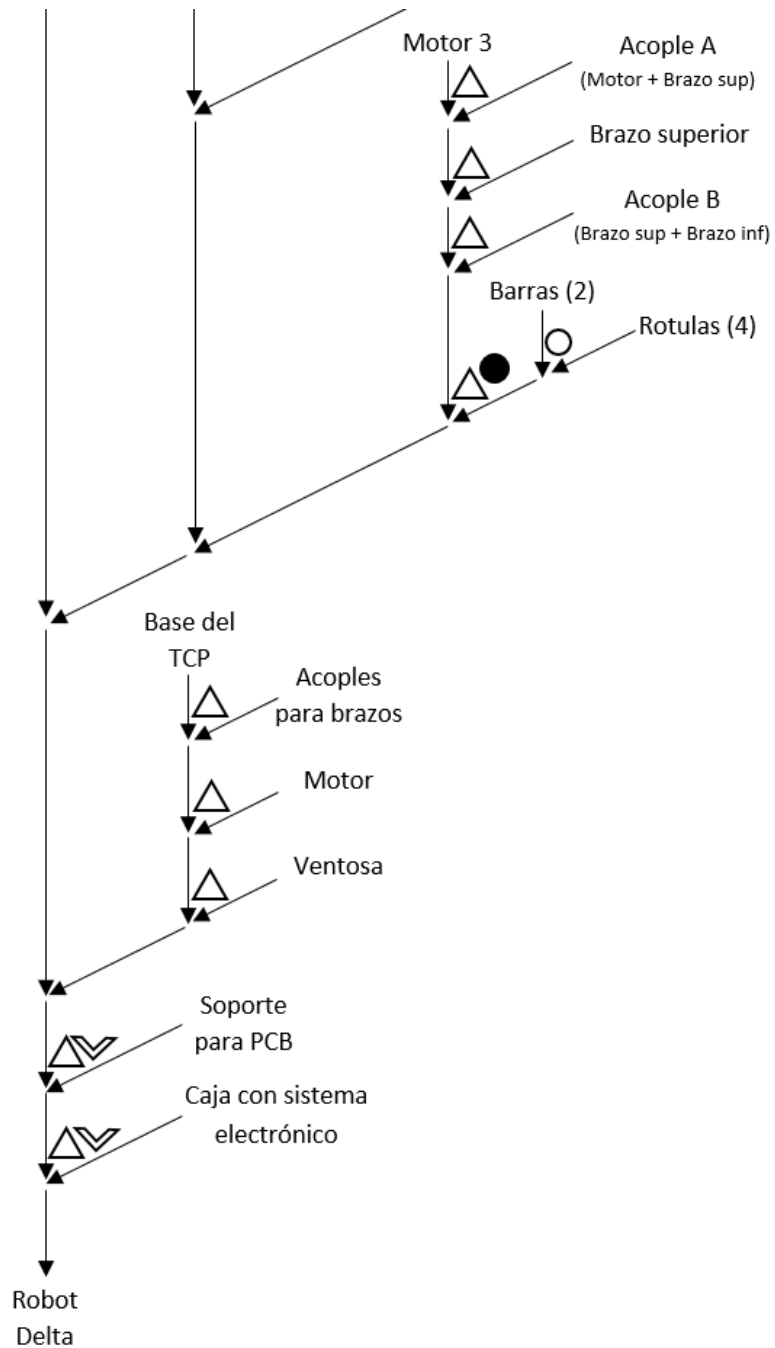


Figura 76.(continuación)



Fuente: Elaboración propia.

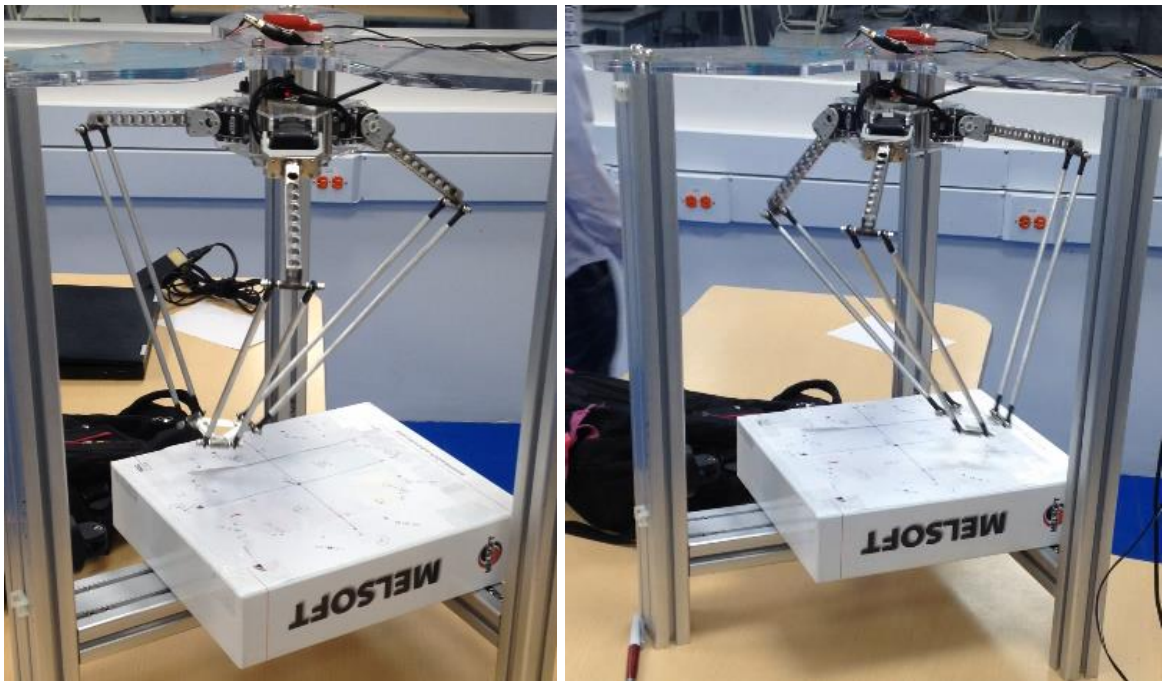
10. RESULTADOS Y DISCUSIÓN

Se realizaron diferentes pruebas que determinaron la validez del diseño, entre las cuales, se consideró el desarrollo de trayectorias sobre el volumen de trabajo establecido. Trayectorias como líneas rectas, círculos sobre los planos XY, YZ y XZ, circunferencias y movimientos programados, fueron distintos patrones que se ejecutaron con el objetivo de analizar la calidad en los trayectos ejecutados sobre el volumen de trabajo.

En la Figura 76, se observa el desarrollo de una prueba de movimiento circular sobre el plano XY.

El objetivo de dichas pruebas fue determinar la correspondencia de la cinemática inversa desarrollada en la aplicación computacional con la posición conseguida por el efector final.

Figura 77. Prototipo versión 1: Prueba de posición.



Fuente: Elaboración propia.

En las Figuras 72 y 73, se observa el prototipo desarrollado.

Figura 78. Prototipo versión 1: Forma de conexión de motores a los brazos.



Fuente: Elaboración propia.

Figura 79. Prototipo versión 1: Ensamblaje de robot en soporte.



Fuente: Elaboración propia.

A continuación, se observa la conexión del motor al brazo, mediante acople mecánico directo (sin reducción). El anexo F ilustra el detalle del acople en mención.

Se decidió que el desarrollo del prototipo de los brazos fuese en acero inoxidable y perfil cuadrado, debido a la rigidez, practicidad y facilidad en la manufactura del mismo. La conexión al acople se realizó mediante lámina metálica de acero inoxidable, acoplado con soldadura MIG (GMAW).³⁴

Figura 80. Prototipo versión 1: Adaptación de motores a brazo.



Fuente: Elaboración propia.

³⁴ Fundamentos de la Soldadura MIG-MAG[En línea] Ingemecánica,, 2015. [Consultado mayo 2016] Disponible en internet: <http://ingemecanica.com/tutorialsemanal/tutorialn53.html>

La Figura 80 ilustra la versión 1 del prototipo, visto desde una proyección frontal-superior. El controlador, aunque no se observa, se encuentra en la zona central de conexión de los tres motores.

Figura 81. Prototipo versión 1: Vista frontal del robot.



Fuente: Elaboración propia.

11. CONCLUSIONES

El presente documento expone el diseño de un robot que permite posicionar componentes electrónicos de montaje superficial, dentro de circuitos impresos, de manera automática, rápida y precisa, mediante un robot paralelo de morfología delta de cuatro grados de libertad, distribuidos en tres destinados para la traslación del efector final y uno para la rotación de los componentes electrónicos.

El robot Delta diseñado permite conseguir velocidades muy altas debido a la manera como los motores, mediante cambios angulares inferiores a los 180° , ejecutan traslaciones sobre todo el volumen de trabajo definido.

El sistema mecánico desarrollado es capaz de soportar los brazos articulados, permitiendo que el efector final pueda realizar las traslaciones necesarias para recorrer el área de trabajo establecido, sin llegar a zonas donde el movimiento se vea restringido mecánicamente.

Los cuatro actuadores, correspondientes a los tres motores conectados a los brazos, y el motor que da rotación a la ventosa, son controlados de manera efectiva mediante el sistema electrónico, cuyo elemento central es un controlador embebido de 32bits, en donde se ejecutan las acciones de control de manera sincrónica y en tiempo real con el software ejecutado desde un computador. Esto permite que el controlador electrónico no realice cálculos complejos (correspondientes a la cinemática inversa), por lo cual, la respuesta del sistema es altamente veloz.

La generación de las trayectorias es efectiva y los movimientos conseguidos por el efector final son suaves, a velocidades variables y sin picos en aceleración debido al sistema de control proporcional desarrollado. Toda la estrategia de control (desarrollo cinemático más generación de trayectorias suavizadas) se ejecuta desde el computador y es enviado al sistema electrónico mediante puerto USB. Esto muestra la efectividad del diseño electrónico y de software.

El software desarrollado en Java responde favorablemente ante la solución de las ecuaciones de la cinemática del robot, permitiendo capturar los datos de coordenadas de los archivos de diseño electrónico, realizando su descomposición

y ejecutando las trayectorias debidas y realizando la transmisión de la información del computador al sistema embebido de manera eficiente sin pérdidas en información.

El desarrollo del prototipo permitió aproximar el diseño a una concepción real, donde las perturbaciones del ambiente pusieron a prueba el control del robot, sin embargo, la respuesta obtenida fue satisfactoria, pues los movimientos conseguidos bajo las trayectorias generadas fueron coherentes y correspondieron al desarrollo cinemático.

Una de las dificultades del ensamblaje de circuitos impresos mediante el presente robot, estuvo en la gran variedad de tipos de componentes electrónicos que existen, por tanto, el ensamblaje se limita a una reducida variedad de dispositivos.

12. RECOMENDACIONES

Se podría considerar una mejoría cambiando la filosofía que se planteó para la comunicación entre la aplicación informática y el sistema electrónico. El modo de conexión y transferencia de la información desde el sistema informático hacia el sistema electrónico se realiza en línea (online), ejecutando el desarrollo cinemático y llevando los datos de posición hacia los motores en pasos seguidos y de manera secuencial, es decir, no existe ningún buffer durante el proceso. Esto podría representar inconvenientes debido a que aunque el margen de error en la transmisión serial es bajo, si durante dicho proceso se presentan problemas de ruido electromagnético, la pérdida de información podría causar movimientos indeseables en el TCP. Por lo anterior, se recomienda rediseñar el concepto del flujo de la información, teniendo en cuenta que el sistema electrónico pudiese contar con un espacio de memoria en donde se pudiese almacenar todas las instrucciones, pasos o secuencias de las trayectorias. La desventaja de esto estaría en que se requeriría un sistema embebido de mayor capacidad de procesamiento, sin embargo, su desarrollo podría mejorar la fiabilidad del robot. Una posible solución estaría en implementar lo anterior bajo el sistema de desarrollo Raspberry Pi.³⁵

Otra posible mejoría podría encontrarse en el cambio de los motores articulares. El robot cuenta con servomotores tipo *Smart* marca Robotis de la línea Dynamixel, los cuales podrían replantearse sobre la línea Dynamixel Pro, los cuales presentan mayor resolución, control y velocidad.

³⁵ Learn and Make with Raspberry Pi, [En línea] Raspberry Pi, Teach, 2016. [Consultado mayo 2016] Disponible en internet: <https://www.raspberrypi.org/>

BIBLIOGRAFÍA

BUDYNAS, Richard, NISBETT, Keith. Shigley's Mechanical Engineering Design. Ninth Edition. New York: McGraw-Hill, 2008. 886. P

BUENO. A. Diseño y Construcción de Circuitos Impresos . [En línea] portaleso. 2015. [Consultado Abril 2016] Disponible en internet: http://www.portaleso.com/portaleso/trabajos/tecnologia/electro/unidad_diseno_y_construccion_de_circuitos_impresos_fotos.pdf.

CAPUZ RIZO, Salvador. Introducción al proyecto de producción. Ingeniería concurrente para el diseño del producto. Valencia: Universidad Politécnica de Valencia. Servicio de publicación, 1999. 104 p.

DEEK, Fadi P.; James A. y McHugh M. Open Source: Technology and Policy. New York: Cambridge University Press, 2009. 1-3 p.

Delta Robots Improve Highly Repetitive Tasks. 2015. Delta Robots Improve Highly Repetitive Tasks. [En línea] robot.fanucamerica. [Consultado Abril 2016] Disponible en internet: http://robot.fanucamerica.com/robotics-articles/delta_robots_improve_highly_repetitive_tasks.aspx.

Generic Requirements for Through-Hole Design and Land Pattern Standard, [En línea] IPC-7251 [Consultado abril 2016]2008. http://www.ipc.org/committee/drafts/1-13_d_7251WD1.pdf

HAGUE R. J. M. y REEVES. P. E., Rapid Prototyping, Tooling and Manufacturing. Reino Unido: Rapra Technology Ltd, 2000. 281 p.

Hand Packing Stations | Pick and Pack Station | Packaging Solutions. 2013. Hand Packing Stations | Pick and Pack Station | Packaging Solutions. [En línea]. combi [Consultado Abril 2016] Disponible en internet: <http://www.combi.com/pick-and-pack>.

LPKF Laser & Electronics North America, ProtoPlace S, [En línea]. lpkfusa 2015. [Consultado Noviembre 2015] Disponible en internet: http://www.lpkfusa.com/products/pcb_prototyping/smt_assembling/protoplace_s/

PANDILOV Z. y RALL K., Parallel Kinematics Machine Tools: History, Present, Macedonia: Mechanical Engineering - Scientific Journal Vol.25, 2006. P.3

PEARCE, Joshua M., Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs. Waltham: Elsevier, 2013,p 5-6

Principios básicos de la robótica. Departamento de Ingeniería de Sistemas y Automática UPV. Biblioteca Universidad Politécnica de Valencia. 2015.

ProtoPlace S, Semi-automatic SMT assembly system [En línea]. lpkfusa. [Consultado Mayo 2016] Disponible en internet: <http://www.lpkfusa.com/SMTAssembly/protoplace.htm>.

R.L. Williams II, The Delta Parallel Robot: Kinematics Solutions, [En línea]. ohio 2016. [Consultado Mayo 2016] Disponible en internet: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf

REDDIT, QM1100 Automatic Pick and Place Machine with Vision system, [En línea]. REDDIT, 2015. [Consultado Mayo 2016] Disponible en internet: https://www.reddit.com/r/PrintedCircuitBoard/comments/358942/qm1100_automatc_pick_and_place_machine_with/

ROJAS, Alvaro Jose. DISEÑO MECATRONICO: Introducción - Proceso de diseño [Diapositivas], Santiago de Cali: Universidad Autónoma de Occidente, 2014.

RUEDA FLOREZ, Jhonnatan Didier.Análisis Estructural de un Robot Paralelo Industrial Tipo Delta. [En línea] Universidad Pontificia Bolivariana2009 [Consultado Abril 2016] Disponible en internet: http://repository.upb.edu.co:8080/jspui/bitstream/123456789/489/1/digital_17613.pdf

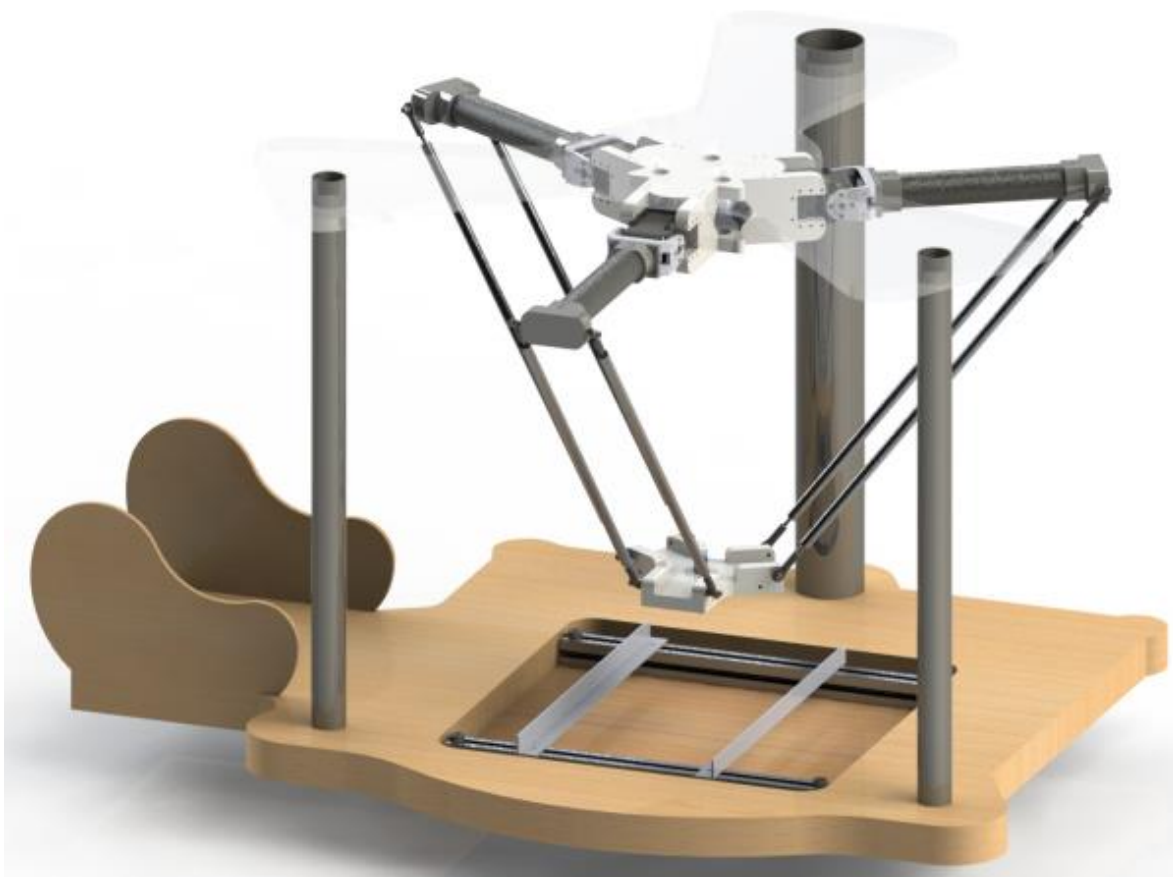
SMTMax, Automatic Pick and Place Machines QM1100-A, 2016. [En línea]. SMTMax, [Consultado Mayo 2016] Disponible en internet: <http://www.smtmax.com/detail.php?id=292>

ULRICH, Karl T., EPPINGER, Steven D., Diseño y desarrollo de productos. 5 ed. México DF: McGraw-Hill, 2012. 150 p.

Universidad Autónoma de Bucaramanga, Cinemática inversa y análisis del espacio de trabajo de una plataforma Stewart, [En línea] UNAB, Vol 1, No 1 2009. [Consultado noviembre 2015] Disponible en internet: <http://revistas.unab.edu.co/index.php?journal=mecatronica&page=article&op=viewArticle&path%5B%5D=1242>

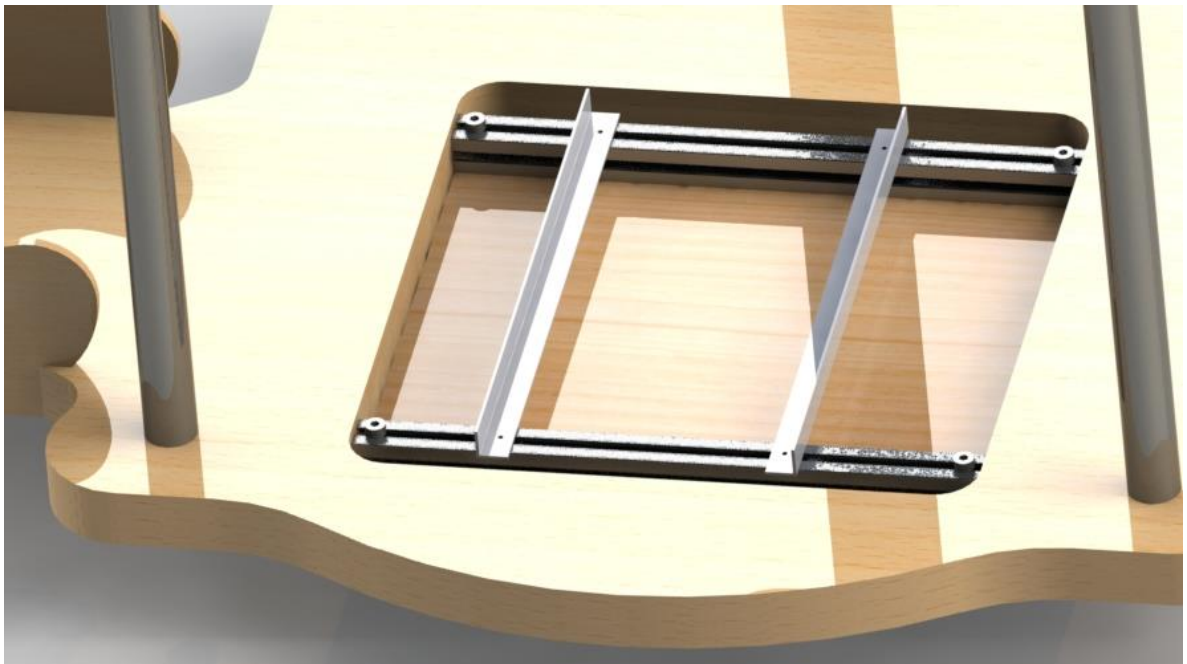
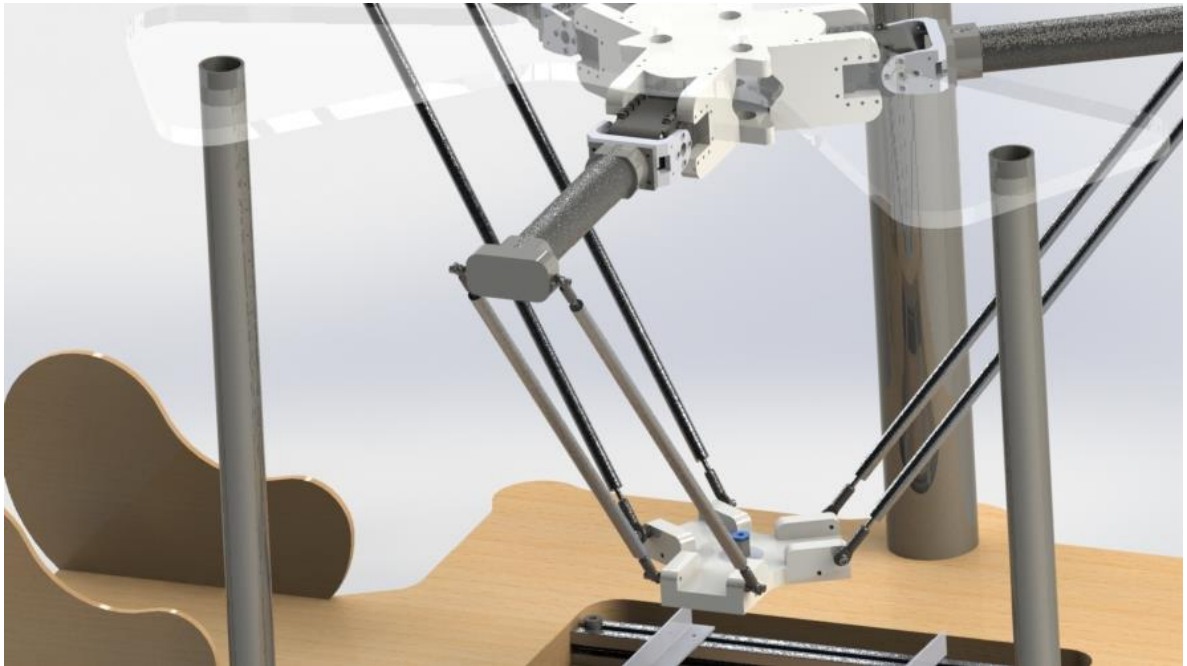
ANEXOS

ANEXO A. RENDER DEL PROTOTIPO FINAL DEL ROBOT.



Fuente: Elaboración propia.

ANEXO B. RENDER DEL DETALLE EN ARTICULACIONES Y SUJECCIÓN DE PCB.



Fuente: Elaboración propia.

ANEXO C. CÓDIGO IMPLEMENTADO EN EL SISTEMA EMBEBIDO ELECTRÓNICO.

```
long data1;
long data2;
long data3;
int sensor;

#define DXL_BUS_SERIAL1 1
#define GOAL_POSITION 30
#define GOAL_SPEED 32
Dynamixel Dxl(DXL_BUS_SERIAL1);

void setup()
{
  pinMode(BOARD_LED_PIN, OUTPUT);
  SerialUSB.attachInterrupt(usbInterrupt);
  pinMode(0, INPUT);
  pinMode(1, OUTPUT);
  // Dynamixel 2.0 Baudrate -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps
  Dxl.begin(3);
  Dxl.jointMode(1);
  Dxl.jointMode(2);
  Dxl.jointMode(3);
  Dxl.writeWord(1, GOAL_SPEED, 1023);
  Dxl.writeWord(2, GOAL_SPEED, 1023);
  Dxl.writeWord(3, GOAL_SPEED, 1023);
}

void usbInterrupt(byte* buffer, byte nCount)
{

```

```

        if(nCount>3)
        {
            toggleLED();
        }
        SerialUSB.print("nCount =");
        SerialUSB.println(nCount);
        data1=0;
        data2=0;
        data3=0;
        for(unsigned int i=0; i < 3;i++)
        {
            data1+=(buffer[i])*pow(10,3-i-1);
        }
        for(unsigned int i=3; i < 6;i++)
        {
            data2+=(buffer[i])*pow(10,6-i-1);
        }
        for(unsigned int i=6; i < 9;i++)
        {
            data3+=(buffer[i])*pow(10,9-i-1);
        }
        //SerialUSB.println("");
        SerialUSB.print("DATA: ");
        SerialUSB.print(data1);
        Dxl.writeWord(1, GOAL_POSITION, data1);
        Dxl.writeWord(2, GOAL_POSITION, data2);
        Dxl.writeWord(3, GOAL_POSITION, data3);
        SerialUSB.println("");
    }

    void loop()

```

```
{  
  sensor = digitalRead(0);  
  if(sensor==1)  
  {  
    SerialUSB.print("sensor");  
  }  
  delay(10);  
}
```

Fuente: Elaboración propia.

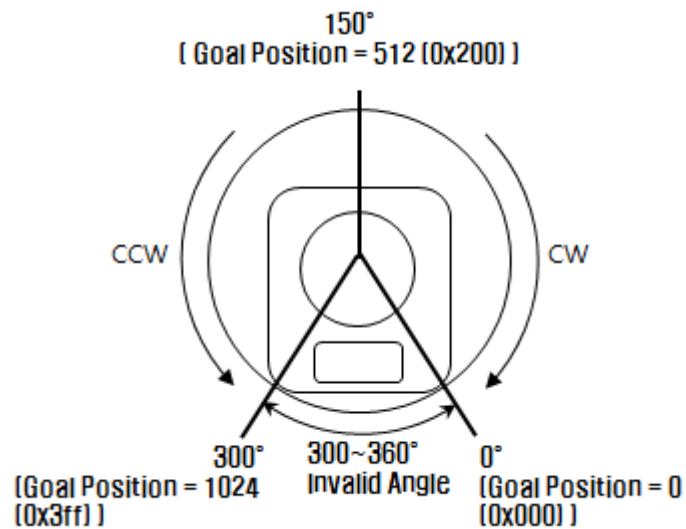
ANEXO D. ESPECIFICACIONES DEL MOTOR SERVOSMART USADOS EN EL ROBOT.

H/W Specification

- Weight : 53,5g (AX-12/AX-12+), 54,6g (AX-12A)
- Dimension : 32mm * 50mm * 40mm
- Resolution : 0,29°
- Gear Reduction Ratio : 254 : 1
- Stall Torque : 1,5N.m (at 12,0V, 1,5A)
- No load speed : 59rpm (at 12V)
- Running Degree
 - 0° ~ 300°
 - Endless Turn
- Running Temperature : -5°C ~ +70°C
- Voltage : 9 ~ 12V (Recommended Voltage 11,1V)
- Command Signal : Digital Packet
- Protocol Type : Half duplex Asynchronous Serial Communication (8bit,1stop,No Parity)
- Link (Physical) : TTL Level Multi Drop (daisy chain type Connector)
- ID : 254 ID (0~253)
- Communication Speed : 7343bps ~ 1 Mbps
- Feedback : Position, Temperature, Load, Input Voltage, etc.
- Material : Engineering Plastic

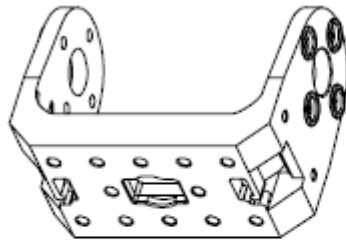
Fuente: ROBOTIS, AX-12/ AX-12+/ AX-12A, 2016. [En línea]. [Consultado Marzo 2016] Disponible en internet: http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm

ANEXO E. RANGO DE OPERACIÓN DEL MOTOR SERVOSMART.

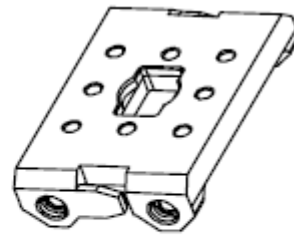


Fuente: ROBOTIS, AX-12/ AX-12+/ AX-12A, 2016. [En línea]. [Consultado Marzo 2016]
Disponble en internet:
http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm

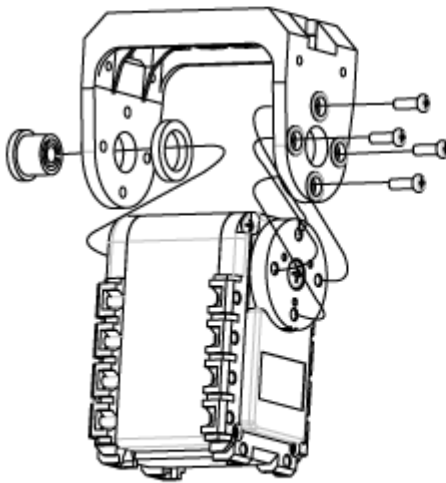
ANEXO F. DETALLE DE PIEZA CONECTORA ENTRE MOTOR Y BRAZOS.



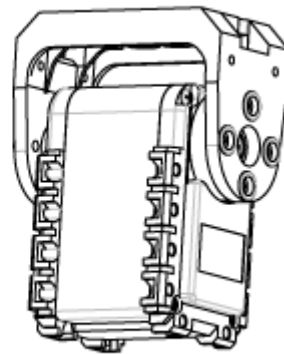
FP04-F2



FP04-F3



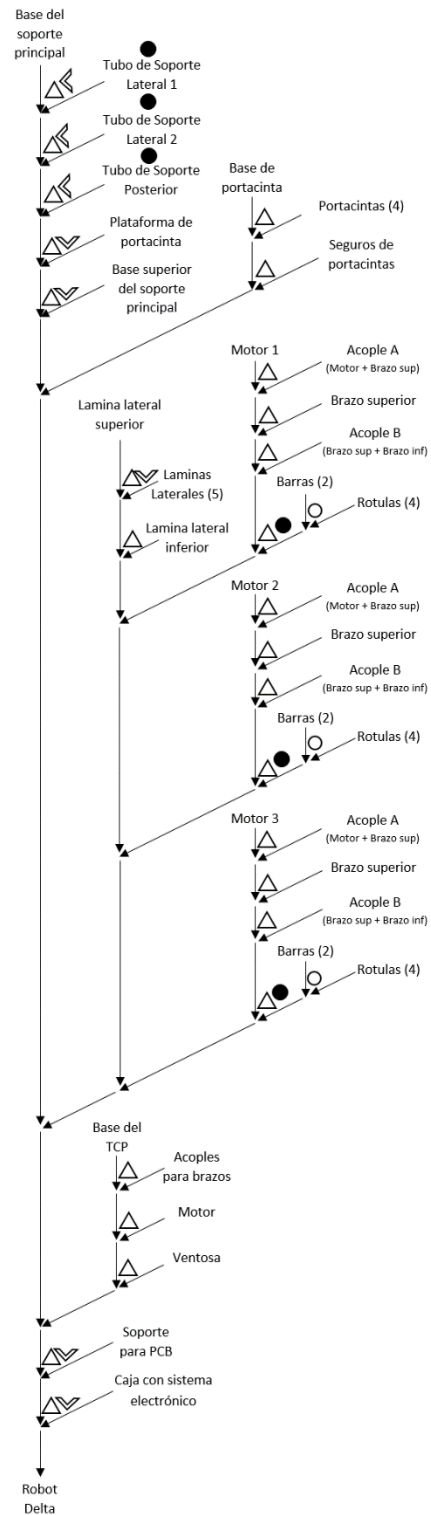
Exploded view



Assembled

Fuente: ROBOTIS, AX-12/ AX-12+/ AX-12A, 2016. [En línea]. [Consultado Marzo 2016] Disponible en internet: http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel/ax_series/dxl_ax_actuator.htm

ANEXO G. DIAGRAMA DE ENSAMBLE



Fuente: Elaboración propia.

ANEXO H. SIMULACIÓN DE MOVIMIENTOS DEL ROBOT MEDIANTE MATLAB.

```
function delta

%% INICIALIZACIÓN

clear;

clf;

clc;

%% ESPACIO GRÁFICO

v = 1500;

axis([-400 400 -400 400 -650 50]);

figure(gcf);

xlabel('X');

ylabel('Y');

zlabel('Z');

set(gca, 'drawmode', 'fast');

view(0,0)

grid;

%% CONSTANTES DEL ROBOT

%datos de la base del robot

sB=350;%567

uB=sB/(2*sin(pi/3))

wB=sB/(2*tan(pi/3))

%datos de el TCP del robot

sP=90;

uP=sP/(2*sin(pi/3))

wP=sP/(2*tan(pi/3))

%datos de las articulaciones

L=192;%524

l=368;%1244

A=60;%131;

%Colores

C= [0 0 1];

C2=[1 0 0];

%% SP

x=0;

y=0;
```

```

z=-300;

%% FUNCIÓN QUE DESARROLLA LA CINEMÁTICA DIRECTA
function pos=fpkDelta(th1,th2,th3)

    SC0=eye(4);

    T0=[eye(3) [0;-L*cos(th1)-wB;-L*sin(th1)];0 0 0 1];

    SC1=T0*SC0;

    Ang0=(0.5*(L*cos(th2)+wB)+L*cos(th1)+wB)/(sqrt(3/2)*(L*cos(th2)+wB));

    T1=[sin(Ang0) cos(Ang0) 0 0;cos(Ang0) -sin(Ang0) 0 0;0 0 1 0;0 0 0 1];

    SC2=T1*T0*SC0;

    SC0=inv(T1*T0)*SC2;

end

%% FUNCION QUE DESARROLLA LA CINEMÁTICA INVERSA
function sp=ipkDelta(x,y,z)

    a=wB-uP;

    b=(sP/2)-(sqrt(3)/2)*wB;

    c=wP-0.5*wB;

    E1=2*L*(y+a);

    F1=2*z*L;

    G1=x^2+y^2+z^2+a^2+L^2+2*y*a-1^2;

    E2=-L*(sqrt(3)*(x+b)+y+c);

    F2=2*z*L;

    G2=x^2+y^2+z^2+b^2+c^2+L^2+2*(x*b+y*c)-1^2;

    E3=L*(sqrt(3)*(x-b)-y-c);

    F3=2*z*L;

    G3=x^2+y^2+z^2+b^2+c^2+L^2+2*(-x*b+y*c)-1^2;

    function qf=ti(E,F,G)

        ta=-(F+(sqrt(E^2+F^2-G^2)))/(G-E);

        tb=-(F-(sqrt(E^2+F^2-G^2)))/(G-E);

        qf=ta;

    end

    theta1=2*atan(ti(E1,F1,G1));

    theta2=2*atan(ti(E2,F2,G2));

```

```

        theta3=2*atan(ti(E3,F3,G3));

        sp=[theta1*180/pi theta2*180/pi theta3*180/pi]';

    end

%% PUNTOS SUPERIORES DEL ROBOT

    %PUNTOS DE LA BASE

    P1B=[eye(3) [-sB/2;-wB;0];0 0 0 1];
    P2B=[eye(3) [sB/2;-wB;0];0 0 0 1];
    P3B=[eye(3) [0;uB;0];0 0 0 1];

    %PUNTOS DE LOS MOTORES

    B1=[eye(3) [0;-wB;0];0 0 0 1];
    B2=[eye(3) [wB*sin(pi/3);wB*cos(pi/3);0];0 0 0 1];
    B3=[eye(3) [-wB*sin(pi/3);wB*cos(pi/3);0];0 0 0 1];

%% OBTENCIÓN DE ÁNGULOS MEDIANTE IPK

    x=0;y=0;z=-200;

    qi=ipkDelta(x,y,z)

    lDelta=dibujaRobot(qi);

    pause(1);

    Tr=genTray2([x,y,z],[124,-75,-350],1);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[124,-75,-400],2);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[124,-75,-350],2);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[-124,75,-350],2);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[-124,75,-400],2);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[-124,75,-350],2);
    moverRobot(Tr,lDelta);

    Tr=genTray2([x,y,z],[0,0,-282],1);
    moverRobot(Tr,lDelta);

%% FUNCIÓN QUE OBTIENE LOS PUNTOS INFERIORES DEL ROBOT

function lDelta=dibujaRobot(q)

    THETA1=q(1);

    THETA2=q(2);

    THETA3=q(3);

```

```

%PUNTO A1
A1=B1;
A1=rot(A1,'X',-90);
A1=rot(A1,'Y',90);
A1=rot(A1,'Z',THETA1);
A1=tra(A1,[L 0 0]);

%PUNTO A2
A2=B2;
A2=rot(A2,'X',-90);
A2=rot(A2,'Y',-30);
A2=rot(A2,'Z',THETA2);
A2=tra(A2,[L 0 0]);

%PUNTO A3
A3=B3;
A3=rot(A3,'X',-90);
A3=rot(A3,'Y',-150);
A3=rot(A3,'Z',THETA3);
A3=tra(A3,[L 0 0]);

%PUNTOS DEL TCP
P1=[eye(3) [x;y-uP;z];0 0 0 1];
P2=[eye(3) [x+0.5*sP;y+wP;z];0 0 0 1];
P3=[eye(3) [x-0.5*sP;y+wP;z];0 0 0 1];

%PUNTOS PARALELOGRAMO A1
A1A=tra(A1,[0 0 -A/2]);
A1B=tra(A1,[0 0 +A/2]);

%PUNTOS PARALELOGRAMO A2
A2A=tra(A2,[0 0 -A/2]);
A2B=tra(A2,[0 0 +A/2]);

%PUNTOS PARALELOGRAMO A3
A3A=tra(A3,[0 0 -A/2]);
A3B=tra(A3,[0 0 +A/2]);

%PUNTOS PARALELOGRAMO 11
l1A=tra(P1,[-A/2 0 0]);
l1B=tra(P1,[+A/2 0 0]);

%PUNTOS PARALELOGRAMO 12
l2A=tra(P2,[(cos(pi/3))*(A/2) -sin(pi/3)*(A/2) 0]);

```

```

l2B=tra(P2, [(-cos(pi/3))*(A/2) sin(pi/3)*(A/2) 0]);
%PUNTOS PARALELOGRAMO 13
l3A=tra(P3, [(cos(pi/3))*(A/2) sin(pi/3)*(A/2) 0]);
l3B=tra(P3, [(-cos(pi/3))*(A/2) -sin(pi/3)*(A/2) 0]);
%*****CREACIÓN DE LINEAS*****
%LINEAS DE LA BASE
lbase1=crealin(P1B,P2B,C);
lbase2=crealin(P2B,P3B,C);
lbase3=crealin(P3B,P1B,C);
lbase=[lbase1,lbase2,lbase3];
%LINEAS DEL TCP
ltcp1=crealin(P1,P2,C2);
ltcp2=crealin(P2,P3,C2);
ltcp3=crealin(P3,P1,C2);
ltcp=[ltcp1,ltcp2,ltcp3];
%LINEAS DE ARTICULACIONES SUPERIORES
las1=crealin(A1,B1,C);
las2=crealin(A2,B2,C);
las3=crealin(A3,B3,C);
las=[las1,las2,las3];
%LINEAS DE ARTICULACIONES INFERIORES
lai1=crealin(A1A,l1A,C);
lai2=crealin(A1B,l1B,C);
lai3=crealin(A2A,l2A,C);
lai4=crealin(A2B,l2B,C);
lai5=crealin(A3A,l3A,C);
lai6=crealin(A3B,l3B,C);
lai=[lai1,lai2,lai3,lai4,lai5,lai6];
%LINEAS DE PARALELOGRAMOS SUPERIORES
lps1=crealin(A1A,A1B,C);
lps2=crealin(A2A,A2B,C);
lps3=crealin(A3A,A3B,C);
lps=[lps1,lps2,lps3];
%LINEAS DE PARALELOGRAMOS INFERIORES
lpi1=crealin(l1A,l1B,C);
lpi2=crealin(l2A,l2B,C);

```

```

    lpi3=crealin(l3A,l3B,C);
    lpi=[lpi1,lpi2,lpi3];
    lDelta=[lbase,ltcp,las,lai,lps,lpi];
end

%% FUNCIÓN QUE MODIFICA EL ROBOT
function modificaRobot(q,lineas)

    THETA1=q(1);
    THETA2=q(2);
    THETA3=q(3);

    %PUNTO A1
    A1=B1;
    A1=rot(A1,'X',-90);
    A1=rot(A1,'Y',90);
    A1=rot(A1,'Z',THETA1);
    A1=tra(A1,[L 0 0]);

    %PUNTO A2
    A2=B2;
    A2=rot(A2,'X',-90);
    A2=rot(A2,'Y',-30);
    A2=rot(A2,'Z',THETA2);
    A2=tra(A2,[L 0 0]);

    %PUNTO A3
    A3=B3;
    A3=rot(A3,'X',-90);
    A3=rot(A3,'Y',-150);
    A3=rot(A3,'Z',THETA3);
    A3=tra(A3,[L 0 0]);

    %PUNTOS DEL TCP
    P1=[eye(3) [x;y-uP;z];0 0 0 1];
    P2=[eye(3) [x+0.5*sP;y+wP;z];0 0 0 1];
    P3=[eye(3) [x-0.5*sP;y+wP;z];0 0 0 1];

    %PUNTOS PARALELOGRAMO A1
    A1A=tra(A1,[0 0 -A/2]);
    A1B=tra(A1,[0 0 +A/2]);

    %PUNTOS PARALELOGRAMO A2
    A2A=tra(A2,[0 0 -A/2]);

```



```

A2B=tra(A2,[0 0 +A/2]);

%PUNTOS PARALELOGRAMO A3
A3A=tra(A3,[0 0 -A/2]);
A3B=tra(A3,[0 0 +A/2]);

%PUNTOS PARALELOGRAMO 11
11A=tra(P1,[-A/2 0 0]);
11B=tra(P1,[+A/2 0 0]);

%PUNTOS PARALELOGRAMO 12
12A=tra(P2,[(cos(pi/3))*(A/2) -sin(pi/3)*(A/2) 0]);
12B=tra(P2,[-cos(pi/3)*(A/2) sin(pi/3)*(A/2) 0]);

%PUNTOS PARALELOGRAMO 13
13A=tra(P3,[(cos(pi/3))*(A/2) sin(pi/3)*(A/2) 0]);
13B=tra(P3,[-cos(pi/3)*(A/2) -sin(pi/3)*(A/2) 0]);

%LINEAS TCP
modlin(lineas(4),P1,P2);
modlin(lineas(5),P2,P3);
modlin(lineas(6),P3,P1);

%LINEAS ARTICULACIONES SUPERIORES
modlin(lineas(7),A1,B1);
modlin(lineas(8),A2,B2);
modlin(lineas(9),A3,B3);

%LINEAS ARTICULACIONES INFERIORES
modlin(lineas(10),A1A,11A);
modlin(lineas(11),A1B,11B);
modlin(lineas(12),A2A,12A);
modlin(lineas(13),A2B,12B);
modlin(lineas(14),A3A,13A);
modlin(lineas(15),A3B,13B);

%LINEAS PARALELOGRAMOS SUPERIORES
modlin(lineas(16),A1A,A1B);
modlin(lineas(17),A2A,A2B);
modlin(lineas(18),A3A,A3B);

%LINEAS PARALELOGRAMOS INFERIORES
modlin(lineas(19),11A,11B);
modlin(lineas(20),12A,12B);

```

```

        modlin(lineas(21),l3A,l3B);
    end

%% FUNCION PARA LA CREACIÓN DE LINEAS
    function linea=crealin(P1,P2,color)

        X=[P1(1,4);P2(1,4)];
        Y=[P1(2,4);P2(2,4)];
        Z=[P1(3,4);P2(3,4)];

linea=line('xdata',X,'ydata',Y,'zdata',Z,'color',color,'erasemode','xor','linewidth',1.5);
    end

%% FUNCION PARA LA MODIFICACIÓN DE LINEAS
    function modlin(linea,P1,P2)

        X=[P1(1,4);P2(1,4)];
        Y=[P1(2,4);P2(2,4)];
        Z=[P1(3,4);P2(3,4)];

        set(linea, 'xdata',X,'ydata',Y,'zdata',Z)
    end

%% FUNCIÓN PARA ROTACIÓN
    function matOut=rot(matIn,eje,a)

        a=(a*pi)/180;

        if(eje=='X')

            matOut=matIn*[1 0 0 0;0 cos(a) -sin(a) 0;0 sin(a) cos(a) 0;0 0 0 1];

        end

        if(eje=='Y')

            matOut=matIn*[cos(a) 0 sin(a) 0;0 1 0 0;-sin(a) 0 cos(a) 0;0 0 0 1];

        end

        if(eje=='Z')

            matOut=matIn*[cos(a) -sin(a) 0 0;sin(a) cos(a) 0 0;0 0 1 0;0 0 0 1];

        end

    end

%% FUNCIÓN PARA TRASLACIÓN
    function matOut=tra(matIn,a)

        matOut=matIn*[eye(3) [a(1) a(2) a(3)]';0 0 0 1];

    end

%% GENERADOR DE TRAYECTORIAS (ESPACIO DE LAS ARTICULACIONES)
    function Tray=genTray(qi,qf,tf)

        N=length(qi);

```

```

Tray=[];
for i=1:N
    a(i)=2*(qi(i)-qf(i))/tf^3;
    b(i)=(-3*a(i)*tf)/2;
    c(i)=0;
    d(i)=qi(i);
end
for t=0:0.1:tf
    for i=1:N
        q(i)=a(i)*t^3 + b(i)*t^2 + c(i)*t + d(i);
    end
    Tray=[Tray;q];
end
end

%% GENERADOR DE TRAYECTORIAS (ESPACIO CARTESIANO)
function Tray=genTray2(pi,pf,tf)
    N=length(pi);
    Tray=[];
    for i=1:N
        a(i)=2*(pi(i)-pf(i))/tf^3;
        b(i)=(-3*a(i)*tf)/2;
        c(i)=0;
        d(i)=pi(i);
    end
    for t=0:0.1:tf
        for i=1:N
            p(i)=a(i)*t^3 + b(i)*t^2 + c(i)*t + d(i);
        end
        Tray=[Tray;p];
    end
end

%% FUNCION PARA MOVEL EL ROBOT
function moverRobot(Tray,lineas)
    Pasos=length(Tray)
    for j=1:Pasos
        sp=Tray(j,:);
    end
end

```

```
x=sp(1);y=sp(2);z=sp(3);  
qf=ipkDelta(x,y,z)  
modificaRobot(qf,lineas);  
pause(0.01);  
end  
end  
end
```

Fuente: Elaboración propia.